NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

*Technical Memorandum No. 33-307*
*Revision 1*

# *Structural Analysis and Matrix Interpretive System (SAMIS) Program Report*

*Robert J. Melosh, Philip A. Diether, and Mary Brennan*
*Philco Corporation, A Subsidiary of the Ford Motor Company*
*Western Development Laboratories*
*Palo Alto, California*



JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA

December 15, 1966

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Technical Memorandum No. 33-307

Revision 1

# Structural Analysis and Matrix Interpretive System (SAMIS) Program Report

Robert J. Melosh, Philip A. Diether, and Mary Brennan

Philco Corporation, A Subsidiary of Ford Motor Company
Western Development Laboratories
Palo Alto, California

Approved by:

M. E. Alper

M. E. Alper, Manager
Applied Mechanics Section
Jet Propulsion Laboratory

JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA

December 15, 1966

# FOREWORD

The work reflected in this report was performed by the Philco Corporation Western Development Laboratories, Palo Alto, California, for the Jet Propulsion Laboratory (under Contract No. 950321), sponsored by the National Aeronautics and Space Administration under Contract No. NAS 7-100.

The research was conducted from February 1963 through December 1965 by members of the Engineering Mechanics Section of the Western Development Laboratories as part of a continuing effort to develop an automated capability for solving problems in engineering mechanics. Contractor personnel were P. R. Cobb (Manager, Antenna Systems Department, Philco), Project Manager; and R. J. Melosh (Manager, Engineering Mechanics), Project Engineer. The principal contractor engineers were H. N. Christiansen and P. A. Diether; M. J. Brennan was senior programmer.

The Jet Propulsion Project Engineers were T. E. Lang (Manager), L. W. Schmele, S. Utku, V. C. Smith, and R. E. Reed.

PRECEDING PAGE BLANK NOT FILMED.



CONTENTS

TABLES

TABLES (Continued)



FIGURES

## INDEX OF USAGE WRITEUPS

INDEX OF USAGE WRITEUPS (Continued)

## ABSTRACT

This report describes the computer aspects of the Structural Analysis and Matrix Interpretive System (SAMIS). It includes a general description of system components and their function, operational, flow, and program intelligence. It describes data and program identification, formats and handling. It defines system tape and core assignments. It includes writeups of each of the subprograms and discusses requirements for preparing input data.

The theoretical basis for the program is contained in a companion report entitled "Structural Analysis and Matrix Interpretive System - Technical Report".

# SECTION 1

# INTRODUCTION

The high speed digital computer has been an increasingly popular tool for structural analysis. A multitude of programs has been generated to handle small-deflection behavior prediction for specialized classes of structures such as trusses, frames, and shells of symmetric forms and loading. Many of these programs are highly efficient packages in terms of economically producing the desired results. On the other hand, because they are specialized, the analyst must become familiar with a vast array of them, or write his own program, unless the scope of his work is parochial. The one or two programs which exist which are sufficiently general have limited flexibility and treat problems of limited size. The first restriction is important because it limits the use of the program and complicates data handling. The second restriction is significant because the persistent trend in computer application is to continually increase problem sizes because cost per calculations are decreasing and better analyses become justifiable.

The Structural Analysis and Matrix Interpretative System (SAMIS) is being developed to simplify automated structural analysis and to eliminate reprogramming for problem changes. These two objectives are, to some extent, in conflict. They are reconciled by choosing the following programming concepts:

1. Wherever possible, standardize. This is achieved in the program for output formats, most input formats, error handling, tape handling and formats, and storage assignment.

2. Provide a modular program. This is achieved by dividing the calculation into a number of tasks which can be performed in the sequence specified by the analyst. A single intelligence system is used. This approach facilitates adding or removing modules from the system.

3. Program for intermediate size problems. This is achieved by providing for the use of tapes for data and program storage. To make this operation efficient, tape search is avoided wherever possible.

This report describes in general and in detail how calculations are directed and how data must be prepared for SAMIS. Material is organized into eight sections. In the next section a general description of the system, its components and flow are included. The third section discusses the program data and program handling. The fourth section describes calculation control by pseudo instructions. The fifth section describes the structural modeling links which, because of their number and size are included separately. The sixth section describes the operation links, their data formats and usage. The seventh section discusses input data form and preparation. The final section includes program flow charts.

# SECTION 2

# STRUCTURAL ANALYSIS AND MATRIX INTERPRETIVE SYSTEM CHARACTERISTICS

This section includes descriptions of the scope of the program, basic concepts on which the program is based, and program and data handling details.

The primary objective of this program is to automate linear structural analysis of shells and provide matrix manipulation by an interpretive program. Shell analysis includes prediction of deflections and stresses of structures under pressure, heat, acceleration, and static loads. In addition resonant frequencies can be obtained. Materials must have linear stress-strain relations, deflections must be such that load-deflection relations are linear. Matrix manipulative capability includes the ability to perform conventional matrix operations; i.e., addition, subtraction, inversion, transposition, eigenvalue and eigenvector determination.

This program can handle matrices of intermediate size (500-10,000) efficiently as well as smaller sizes (less than 500) with a small penalty as compared to a specialized in-core routine. Whenever possible problems are kept in core to avoid the use of tape storage which is inherently more time consuming. Answers are obtained in a single pass on the machine. Emphasis has been placed on self-checking, simple recovery from faults, and preservation of calculations up to the fault point. Application flexibility is attained by leaving operation sequence and data disposition under control of the analyst.

## 2.1 SYSTEM COMPONENTS

The system is composed of four components: The initiating link, the Master Intelligence, the operation links, and the input data. The initiating link (MAKER) generates the problem flow control data. Master Intelligence (MINTS) provides the link and data handling intelligence. MINTS subroutines control tape search, tape reading, matrix coding, and error operations. The operation links perform operations on arrays of data. Matrix sorting, addition, inversion, multiplication, input and output handling are some examples of operations performed. Input data are required by the links to define the operations to be performed (pseudo instructions) and data for the particular problem under consideration. The pseudo instructions define data assignments, data tape assignments, matrix identification and the sequence of link selection.

## 2.2 SYSTEM FLOW

Initially, the MAKER link reads a set of pseudo instructions and generating a pseudo program from the pseudo instructions. The pseudo program tape is written and rewound, then the pseudo instructions are read into core one at a time. For each instruction, MINTS locates the required data, assigns core storage areas, reads the subroutine if required, and transfers control to the subroutine. Upon return from the subroutine, MINTS provides error diagnostics and then reads and executes the next pseudo instruction.

As an example, the steps taken by the computer are defined in Table 2-1 for generating two component stiffness matrices and summing them. This problem includes all the steps characteristic of a more complex problem involving other subprograms and subroutines. All steps are independent of matrix size.

TABLE 2-1

EXAMPLE PROBLEM

| Step Number | Operation |
|---|---|
| 1 | Read the pseudo instructions and generate (MAKER) the program |
| 2 | Read a generated pseudo instruction |
| 3 | Select the BILD program from tape and read into core |
| 4 | Transfer control to the BILD package |
| 5 | Generate matrices and write on tape |
| 6 | Return Control to the Master Program Control |
| 7 | Read the addition (ADDS) pseudo instruction |
| 8 | Locate the appropriate matrices from core or tape |
| 9 | Select the ADDS program from tape and read into core |
| 10 | Enter the subprogram (ADDS) and add the matrices |
| 11 | Read the HALT pseudo instruction |
| 12 | Halt |

## 2.3 CORE AND TAPE ASSIGNMENTS

Core storage is divided into three basic regions: systems, program, and common. Systems storage contains the IBSYS, MONITOR or other computer job system components necessary for job control.

Programs in storage include the MINTS program or the current link being used for a particular operation. Links are brought into core as they are needed.

Common storage contains the parameters common to most of the links. These one-dimensional arrays are defined in Table 2-2. These arrays make the pseudo instruction, matrix identification and matrix data region accessible to all links and subprograms.

The ABA table defines all acceptable logic pseudo instructions. The ABB table defines all acceptable operation instructions and the designations of the respective operation subprograms. All pseudo instructions are checked against the instructions in these tables. These tables are read from cards by the initiating program, MAKER. Format and content of these data are described in Section 4.

The pseudo instructions region (BUR) contains a current instruction being operated on by the MINTS. The identification region (TRA) defines the A, B, and C matrices being operated with, or, the most recent ones operated upon. The KOD region is used by COINS in the process of coining or decomposing row and column codes. The function of the read (REE) and write (WRY) blocks is to provide a space where data can be stored when it is read to and from tape. These blocks serve as buffers in core for the tape.

TABLE 2-2

COMMON DATA

| Array | Dimension | Location* | Function |
|---|---|---|---|
| ABA | 20 | 1 | Logic pseudo-code table. |
| ABB | 100 | 21 | Operation pseudo-code, link designation, print control, and tape assignment table. |
| BUR | 24 | 121 | Current pseudo-instruction and next pseudo-instruction. |
| NCE | 1 | 145 | Word error indicator. |
| TRA | 60 | 146 | Matrix identification. |
| KOD | 5 | 206 | Region for forming and decomposing codes. |
| NLD | 1 | 211 | Word defining dimension of DAT array. |
| NBD | 1 | 212 | Word defining number of data blocks in DAT. |
| REE | 120 | 213 | Tape reading buffer. |
| WRY | 120 | 333 | Tape writing buffer. |
| DAT | X | 453 | Matrix data and erasable storage. |

* location relative to first assigned common location.

The data region (DAT) is the largest array and encompasses the remaining common storage in core. This region is broken into one, two, or three parts depending upon how many matrices are being handled and the storage assigned by MINTS or the link. In the 32K core, space allocated to this open ended array involves 20,000 locations.

Table 2-3 gives a list of the tape numbers and their assignments. Whenever possible the use of the assigned scratch tapes should be avoided for intermediate calculations since these tapes are used by some links for intermediate data storage. This means that data on these tapes may be erased by a subprogram. The use of matrix tapes should be avoided if possible. The primary reason is because tape usage is a slow process compared to in-core operation. Core operation should always be used if the matrices will fit in core. However, if tapes must be used because of the size of the problem, then as many tapes as possible should be used to minimize tape search time.

Writing and reading of matrix tapes is performed only by the TAPES subroutine. On the IBM 7094-7040 Direct Coupled System, tapes are simulated on storage disks. Then TAPES controls writing onto the disks. Data are assigned to matrix tapes in conformance with the pseudo instructions (Section 4).

## 2.4 DATA HANDLING

The handling of data can be divided into three categories: the handling of input, in-transit, and output data. Input data is always read from Tape 5. The input data can be of the following kinds: pseudo instructions, material tables, element data, matrices and matrix heading cards. Pseudo instructions are always the first data read in for a job. After

TABLE 2-3

SAMIS TAPE ASSIGNMENTS

| Logical | Physical | SAMIS | IBSYS (JPL) |
|---|---|---|---|
| 1 | A1 | Operating System | SYSLB1 |
| 2 | B2 | Debug Library* | SYSUT3 |
| 3 | B3 | Pseudo-program | SYSUT4 |
| 4 | A4 | SAMIS Library* | SYSUT1 |
| 5 | A2 | Card Input | SYSIN1 |
| 6 | A3 | Printed Output | SYSPP1 |
| 7 | B4 | Punch Output | SYSPP1 |
| 8 | B1 | Scratch* | SYSUT2 |
| 9 | A5 | Matrix Data | SYSCK1 |
| 10 | B5 | Matrix Data | SYSCK2 |
| 11 | A6 | Matrix Data | SYSUT5 |
| 12 | B6 | Matrix Data | SYSUT8 |
| 13 | A7 | Matrix Data | SYSUT6 |
| 14 | B7 | Matrix Data | SYSUT9 |
| 15 | A8 | Matrix Data | SYSUT7 |
| 16 | B8 | Scratch* | SYSUT0 |

* These tape assignments are subject to modification to meet installation requirements; see table 4-6.

the pseudo program has been generated it calls in the next three kinds of data as they are needed.

In-transit data includes matrices and links. Matrices can be sent to and from scratch tapes or matrix tapes. The form of matrices sent to or received from scratch tapes depends on the subprogram writing the tapes when these tapes are used as intermediate storage in the link. However, matrices sent to or received from matrix tapes (by TAPES) will be in one of two forms; either coded or pre-coded. Scratch tapes will be written in the same way as matrix tapes if they are designated as output matrix tapes in a pseudo instruction.

In coded format the matrix consists of a one-dimensional array with two alternating terms. The odd numbered terms are called codes. They are packed words defining the row and column numbers of the matrix coefficients. The even numbered terms are the floating point decimal elements. They are the value of the matrix element at the location defined by the preceding code. If a zero code occurs, this indicates that the array is terminated. A coded matrix may be row listed (elements of each row given in increasing column order and then subsequent rows listed in order), column listed or unsorted when input. All coded matrices generated by the links are in row or column listing depending on the link.

In precoded format the matrix is a one-dimensional array consisting of three parts. The parts are the row codes, column codes, and elements. The row code is a packed word containing the row gridpoint number and component number. The column code is similar. The elements are the values of the matrix. The order of the three parts depends on whether the matrix is row listed or column listed. If row listed, the column codes are listed first immediately followed by the row codes. Then, beginning a new block, the matrix elements are listed by rows. If column listed, the row codes, then column codes, and finally elements are listed, the elements being listed by columns.

Links are in-transit data controlled by the CHAIN subroutine which is a part of the job system. CHAIN searches tape and brings links from tape to core. Since links are always read into the program storage area, previous links in core are destroyed when the new link is read.

Output data is in three forms: matrices, error statements and error exit data. Matrices are printed and/or punched in either coded or precoded format by INKS. (Note that punching may be excluded by systems software). "Heading" statements desired by the analyst are printed out with the A1 (first matrix) matrix only. Heading cards are read from the input tape and immediately printed one card at a time. Heading statements except the first are always listed on separate pages from the listing of the matrix. The first heading card appears with the matrix. Their format is 12A6.

Error statements are those error comments printed out by a program. Error comments are printed when an error is encountered. Insofar as possible, all format, listing, identification, and size checks are performed before the error exit is taken from a program.

Error exit data output occurs when a nonrecoverable or flow disrupt error occurs. In these cases the status of the core and tapes is identified by printing the matrix identification data along with the instruction being worked on. Matrix identification includes all pertinent information of the TRA block (See Section 5). The error exit printout is controlled by the subprogram ERROR.

Data and identification information are stored on matrix tapes in blocks of 127 words each. The first 120 words are data and the last seven words are matrix and tape identification information (TRA 1-8).

Each array is assigned a tape group number. Group numbers, which identify the location of matrices on tape, must be in sequential order, e.g., 1, 2, 3, 4 . . . <u>Also, it is not permissible to write a new matrix in the middle of a sequential group</u>. The new matrix may be shorter or longer than the one it is being written over, in which case, the identification associated with the overwritten or succeeding matrix will be partially destroyed, thus interfering with subsequent logical tape search.

Tape search is accomplished by FINDS (pg 33) as follows. The tape backspaces one block, checks the group number, then determines whether to move forward or backward to reach the correct group number. If, after the one backspace, the block read is in the group desired the tape looks at the block number and backs up the absolute value of the number. This positions the tape at the beginning of the group to be read.

2.5 IDENTIFICATION SYSTEMS

Three systems are used to discriminate between subprograms, matrices, and data. Subprograms are identified by a chain number on the program tape. The correspondence between chain numbers and subprogram instructions is contained in COMMON in the ABB table. Further information on chain numbers is given in Section 2.6.

Matrices are identified by two words. The first word consists of three alphabetic characters. The second word consists of three numeric characters, e.g., MAT106. Other identification which characterizes the form of a matrix is stored in the TRA region of common storage and in each block of data on tape. In the TRA region, these data are ten words and have the form and interpretation given in Table 2-4.

From these ten words, seven are formed and stored on tape with each block of data. These are the last seven words in the block. The first word is the matrix alphabetic identification; the second word is the matrix numeric identification; the third, the number of the block of data; the fourth the number of rows and columns in the matrix (a packed word); the fifth, the listing of the data, (row, column, or unsorted); the sixth the format of the coding, (coded or precoded); and the seventh, the tape group number. The purpose of including the last piece of information is to provide the MINTS with the capability of backspacing records on tape and of checking to make certain that the tape record is the one required and in proper sequence.

The data in the matrices are identified by codes. The codes are 36 bit packed words containing column, row, and component data identifying the gridpoints of the structure. The coding system is based on the nodal numbers associated with the matrix during its development and the directional component at that gridpoint. A maximum of 4095 gridpoints are admissible by the coding system although it is possible to handle more gridpoints in pieces. A maximum of ten components of force or displacement may be considered at a gridpoint.

TABLE 2-4

MATRIX IDENTIFICATION

| ITEM | FORM | INTERPRETATION |
|---|---|---|
| 1 | Alphabetic: | Matrix letter designation |
| 2 | Integer: | Matrix numeric designation |
| 3 | Integer: | Number of data blocks in the matrix |
| 4 | Integer: | Number of rows in the matrix (precoded) or maximum number of blocks (coded) |
| 5 | Integer: | Number of columns in the matrix (precoded) |
| 6 | Integer: | Listing indicator -1, row listed; 0, unsorted; +1 column listed |
| 7 | Integer: | Coding format 0, coded; 1, precoded |
| 8 | Integer: | Tape group number |
| 9 | Integer: | Index locating data in DAT. If $>0$, matrix is in core at DAT (index value If $=0$, matrix is not in core and tape is not positioned for reading. If $<0$, matrix is on tape, tape is positioned and matrix can be read into DAT (-index value). |
| 10 | Integer: | Initial TRA index of identification data for a matrix TRA (10) locates A matrix identification, TRA (20), B, and TRA (30), C. |

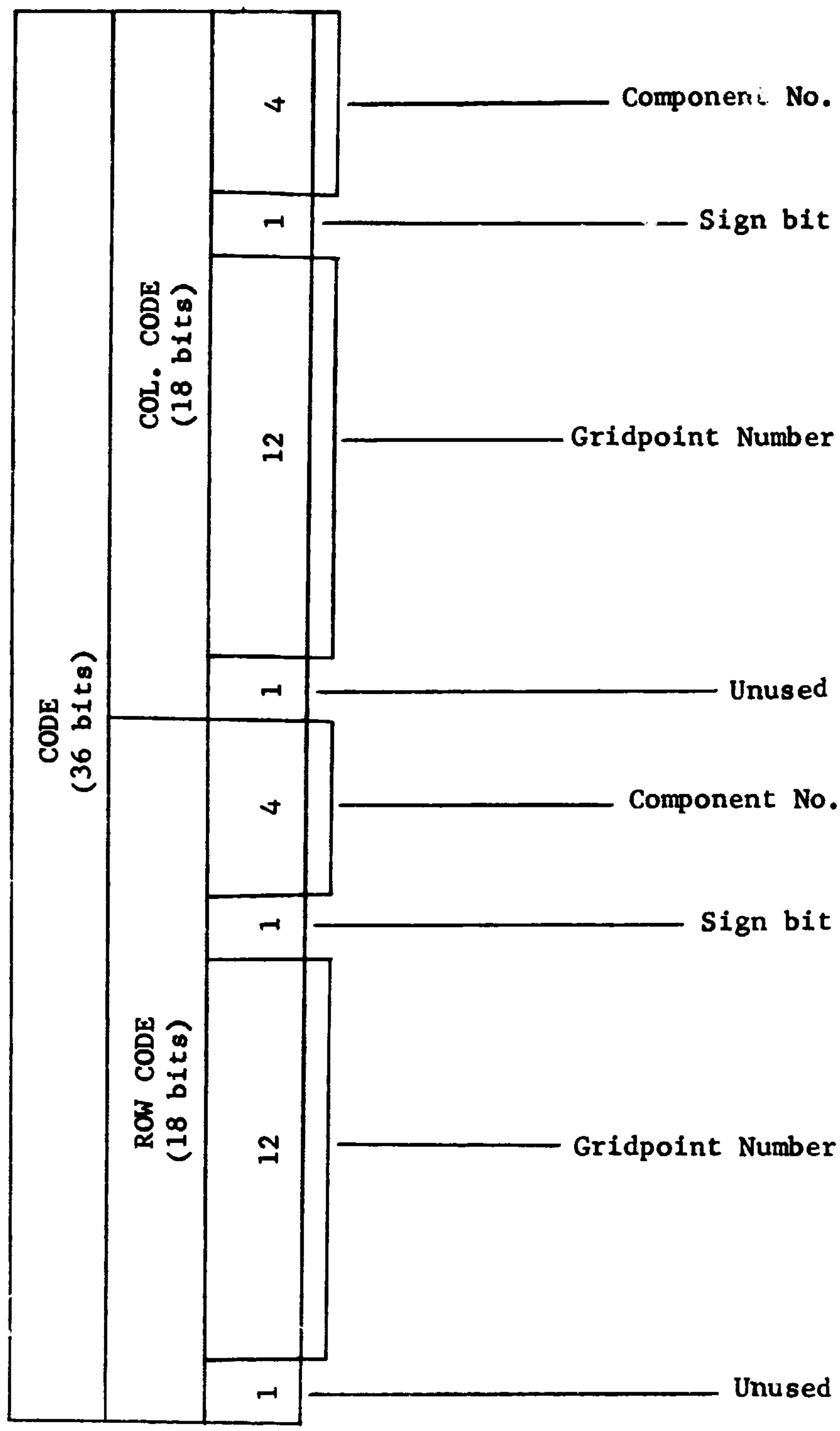
A graphical breakdown of the codes appears in Figure 2-1. Four bits of each half of the word are reserved to define the component at the grid-point and 12 bits to define the gridpoint number The remaining bit (the sign bit) designates the coordinate system used When only the row or column number is required as in the precoded matrices the upper part of the word identifies the row or column and the lower 18 bits are zero.

## 2.6 LINK HANDLING

Links are handled using the IBM CHAIN subroutine. During preload time, the links are stored on the SAMIS program tape. An initiating link (MAKER) calls CHAIN and directs read-in of MINTS.

Upon call for an operation link, the chain table (ABB) is searched to find the instruction name and the link designation which follows it. The link designation includes the chain number of the link. (Interpretation of designation numbers is defined in Section 4.2). Control is then transferred to CHAIN. This subroutine searches the SAMIS tape, reads the required link, and transfers control to it. The exit instruction taken from the link is a call to the CHAIN subroutine.

Link chain numbers are given in Table 2-5. This table also includes the subprogram label used for source and binary decks. Binary decks for each link may contain more than one alphabetic label since many routines are used in more than one link.




CODE WORD BREAKDOWN

Figure 2-1

TABLE 2-5

LINK IDENTIFICATION

| Link | Chain Number | Label |
|---|---|---|
| READ | 1 | A |
| MAKER | 2 | B |
| COLS | 3 | C |
| ROWS | 3 | C |
| FLIP | 5 | E |
| CODE | 6 | F |
| DECO | 7 | G |
| ADDS | 8 | H |
| SUBS | 8 | H |
| MULT | 9 | I |
| WASH | 10 | J |
| ITER | 11 | K |
| CHIN | 12 | L |
| CHOL | 13 | M |
| SORT | 3 | N |
| ROOT | 15 | O |
| BILD | 16 | P |
| INKS | 18 | R |
| MINTS | 26 | Z |

## 2.7 LINK STANDARDS

Standard requirements for links provide for inter-link communication and operation of the links as a system. These include minimum handling capabilities, data checking, and system compatibility requirements. Additional restrictions and capabilities in the links are described in link writeups. It is noted that requirements cited here apply to existing links as well as new links that may be added to SAMIS.

Restrictions on link input and output are as follows:

(1) Data read from cards must be read from tape 5. Card formats, sort, and quantity are independently defineable for each link, but matr .. format is preferable.

(2) Matrix data must be read or written on at most six distinct tapes except that data written on scratch tapes during link execution may be read within that link. Input and output tapes are also available for link use. Matrices should be used from core if available.

(3) Matrix data for input and output must be in either coded or precoded format. Coded matrices need not be sorted. All matrix tape data must be read and written in blocked form using the subroutine TAPES. Coded matrices must fill the last block or be ended by a zero in the last block. The first block must be numbered 1 and succeeding blocks numbered in sequence. The last block must have a negative block number.

(4) All printed link output must be written on tape 6.

Minimum functional requirements of each link are as follows:

(1) Format, listing, quantity and shape characteristics of data must all be checked before an error flag is indicated (NCE = 1) and exit taken from the link. Other errors revealed during calculations require the non-recoverable error flag.

(2) If the program is incapable of handling the volume of data involved, the recoverable error flag is set (NCE = -1) before link exit. If no limits exist in the link, the recoverable flag may be used for some other purpose.

(3) The matrix identification array (TRA) must be appropriately modified before link exit to indicate which matrices are still intact and where.

Link compatibility requirements are as follows:

(1) Upon entry, the COMMON region of core will contain the following one-dimensional arrays:

| | |
|---|---|
| ABA: | 20 possible logic pseudo codes |
| ABB: | 100 possible operation codes |
| BUR: | The 24 words of the current pseudo instruction |
| NCE: | The 1 word error indicator |
| TRA: | The 60 words for the A, B, and C matrix identification data |
| KOD: | The 5 word region for forming and decomposing codes |
| NLD: | One word defining the dimension of DAT |
| NBD: | One word defining the number of blocks in DAT |
| REE: | The 120 word storage region for reading tape blocks |
| WRY: | The 120 word storage region for writing tape blocks |
| DAT: | The region for data storage containing NLD locations |

Of these arrays, ABA, ABB, NLD, and NBD must not be changed by the link. Contents of ABA and ABB are described in paragraph 4.2.

(2) Each link must be given a one word designation denoting its input and output limitations. This designation is used by Master Intelligence and consists of four numbers packed as in Figure 2-1. The first

number (bits 2-13) is the chain identification number; the second (bits 14-18) the identity of input matrices required in core upon link entry; the third (bits 20-31) the identity of input matrices permitted in core; and the fourth (bits 32-36) the number of output matrices required. Input matrices are counted from the left on the pseudo instruction; output matrices from right to left. The identity of input matrices is a binary coded number where bits correspond to the A, B, and C positions respectively. If this is six, for example, the A and B matrices are denoted. If two only, B. This designation appears in the coding information of each link. The subprogram COINS may be used to separate components of this word.

## 2.8 ERROR CORRECTION AND RECOVERY

Two types of error may be encountered in operation within SAMIS. "Non-recoverable" errors cause the computer to print a suitable comment on the output tape and terminate. "Recoverable" errors will involve a diagnostic printout also, but calculations will continue if the analyst has provided for the error.

When the engineer wishes to restart calculations in his problem he can re-enter the machine in the usual way except that required tapes must be re-installed on the computer. Any new instructions that the engineer wishes to insert in his program are inserted and those program instructions that have been successfully performed may be omitted if required data is still available for subsequent calculations. Use of the restart feature allows the engineer to avoid recalculation.

Non-recoverable errors are signalled whenever parameters assume unacceptable values in the analysis. A diagnostic printout is provided from the link and MINTS recalled. Section 2.4 data provides further explanation of error printouts.

Recoverable errors, the second type of errors, are those that the engineer may anticipate. These usually are involved with the fact that the matrix being handled is too large for the program link being used. Suitable comments are written on the output tape and the program will follow a sequence of corrective operations required by the engineer and initiated with an ERRS instruction. These may include use of more capable links or instructions to handle matrices in partitions.

Table 2-6 lists recoverable errors of the operation links. Any given link has at most one recoverable error exit.

In addition to error control within SAMIS, the Fortran system provides error analysis of input-output operations. The most common error involves an error in input card format. If a format error is encountered in a program, control is transferred to the system subroutine EXEM (execution error monitor). This subroutine in turn sets up the required error indicator, prints diagnostic comments, and terminates the job.

Similarly, if SAMIS requests reading of input data beyond that which is available, the monitor will end the job.

## 2.9 OPERATION WITH MONITOR SYSTEMS

Operation of SAMIS under a monitor will vary with the capabilities and options of the monitor. The next paragraphs define operation under the Fortran monitor of the IBSYS Monitor System, Version 13 and the Direct Coupled operating system used with Ames' IBM 7040-7094. It illustrates how efficient SAMIS use can be provided using CHAIN segmentation.

TABLE 2-6

RECOVERABLE ERRORS

| Link | Cause of Error |
|---|---|
| CHIN | A matrix is too large for link |
| CHOL | A matrix is too large for link |
| CODE | Too many codes in the precoded matrix to fit in core |
| COLS | Matrix is not in coded format |
| DECO | Same as above (CODE) |
| INKS | Precoded matrix will not fit in core |
| ITER | Matrix order is too large |
| MINTS | Matrices required to fit in core will not fit |
| MULT | Neither A nor B fit in core |
| ROOT | Matrix order is too large for link |
| ROWS | Matrix is not in coded format |
| WASH | The C matrix does not fit in core and no output tape noted |

Master intelligence checks the size of the matrices for all links to see that matrices required to fit into core actually do fit. If not a recoverable error is signalled.

Decks under this monitor take one of three forms. The Tape Generation form is used in writing a permanent SAMIS library tape. The Production Run form is used for running jobs of five or more minutes using the SAMIS library tape. The Debug Run form is used when some links are to be taken from the library tape and some to be loaded for the job. Note that with the library tape being used (4), it is not possible to compile a source deck and test a link in the same run.

## Generation of a SAMIS Library Tape

A tape containing all checked out links of SAMIS is written to reduce load time on normal runs. Writing of this tape is accomplished by performing the following:

1. Assembling binary cards of all the SAMIS routines required for each LINK. (Routines required are defined in this document in each link writeup). Note that each link is terminated by CALL CHAIN (N, Z), where Z is previously set to ABB(I) and I = 4 for for standard links, 6 for special links, and 8 for MINTS.
2. Insuring the existence of a CLOCK subroutine. This subroutine is called by the MINTS link. Upon subroutine return, MINTS optionally prints the execution time. (This option is controlled by data in the ABB array as described in Table 4-6 and paragraph 3c below).

   If a CLOCK subroutine is not available in the monitor library, a dummy routine is used.

3. Preparing IBSYS and Fortran control cards. These are described under Item 4.

4. Assembling the complete deck. The general form of deck elements is described in Table 2-7. Ordering of links after the MINTS link will affect the amount of tape search time during problem solution. Most frequently used links should be placed first and infrequently used ones last.

5. Warning operators of calculation action and submitting the deck. Tape writing requires about 3-1/2 minutes of computer time.

   After the tape has been written, the Fortran monitor will try to run a CHAIN job. Since the deck and job setup do not comply with the standard chain job, execution will fail. Failure may evoke a dump, despite successful writing of the SAMIS library tape.

Execution of a Production SAMIS Job

During a production run, the MINTS link is put on Tape 2, preceded by a starter link. The starter link merely calls CHAIN (2,4) using the CHAIN subroutine. The MINTS link is called from 2 and all other links are taken from the SAMIS library tape, 4.

Table 2-8 defines the deck form for the production job. Note that the data part of the deck contains the Tables ABA and ABB described in Section 4. The eighth word of ABB is set equal to the sixth word so that MINTS is taken from the special tape. This accelerates tape search for MINTS.

TABLE 2-7

SAMIS LIBRARY TAPE RUN
Generation Deck Form

| Element | Form | Function |
|---|---|---|
| IBSYS Identity Card | $JOBbbbDEBUGbNAME......[1]XXXXXA7045[2]bbb1000[3]bb3.70[4] | Start Job |
| IBSYS Setup Card | $SETUPb4 | Define a Physical Tape |
| IBSYS Monitor Card | $EXECUTEbbbbbbFORTRAN | Call Fortran Monitor |
| FORTRAN Starter | *bbbbbXEQ | Start Execution |
| FORTRAN Count Card | * | Comply with System Limitations |
| FORTRAN Chain Card | *bbbbbCHAIN(26,2) | Mints Chain Card |
| FORTRAN Chain Link | Binary Cards | Mints Chain Link |
| FORTRAN Chain Card<br>FORTRAN Chain Link | Repeated, as above, for each link to be written on the SAMIS library tape. | Other Links |
| FORTRAN Data Card | *bbbbbDATA | Force Writing of SAMIS Tape |

[1] Analyst's name; [2] Job number; [3] Number of lines of printed output; [4] Run time in minutes.

## Execution Of A Debug SAMIS Job

During a debug run, all links but the starter and links being tested are taken from the library tape. Links to be tested are loaded on 2, preceded by the starter link.

Table 2-9 defines the deck form for a debug job. The changes of this deck from the production deck occur in the ABB table entries. In addition, of course, the new links are read in.

Changes in the ABB involve the eighth word of the first card and the chain word for each link being tested. The eighth word of ABB is set to the logical tape number of the SAMIS library so that MINTS is read from that tape. The chain word is increased by $004000000000_8$, over the value assigned for the production ABB (See Table 4-6), for each link to be tested. This effectively increases the link number by 64 so the link is selected from 2. Thus if the WASH link is to be tested, the entry following the word WASH on the fourth ABB card is changed, from $000504000301_8$ to $004504000301_8$. All links to be tested must have their chain word modified in ABB, or that link will be selected from 4.

TABLE 2-8

SAMIS PRODUCTION RUN
Deck Form

| Element | Form | Function |
|---|---|---|
| IBSYS Identity Card | $JOBbbbPRODbbNAME.......[1]XXXXXXA7046[2]bbb0400[3]bb2.00[4] | Start Job |
| IBSYS Setup Card | $SETUPb4NNN[5] | Define SAMIS Library Tape Number |
| IBSYS Monitor Card | $EXECUTEbbbbbbFORTRAN | Call Fortran Monitor |
| FORTRAN Starter | *bbbbbXEQ | Start Execution |
| FORTRAN Count Card | * | Comply with System Limitations |
| FORTRAN Chain Card | *bbbbbCHAIN(21,2) | Starter Link Chain Card |
| FORTRAN Chain Link | Binary Cards | Starter Link |
| FORTRAN Data Card | *bbbbbDATA | Indicates Input Data Start |
| ABA Input | BCD Card (see Table 4-6) | ABA Table |
| ABB Input | BCD and Octal Cards (see Table 4-6) | ABB Table |
| Pseudo Instructions | BCD and Integers | Define Pseudo Program |
| Data | Floating Decimal, Integers, BCD (see Link Writeups) | Define Problem Parameters |

[1] Analyst's name; [2] Job number; [3] Number of lines of printed output; [4] Run time; [5] Tape number of physical tape containing SAMIS library.

TABLE 2-9

SAMIS DEBUG RUN
Deck Form

| Element | Form | Function |
|---|---|---|
| IBSYS Identity Card | $JOBbbbPRODbbNAME.......[1]XXXXXXA7046[2]bbb0400[3]bb2.00[4] | Start Job |
| IBSYS Setup Card | $SETUPbB3NNN[5] | Define SAMIS Library Tape Number |
| IBSYS Monitor Card | $EXECUTEbbbbbbFORTRAN | Call Fortran Monitor |
| FORTRAN Starter | *bbbbbXEQ | Start Execution |
| FORTRAN Count Card | * | Comply with System Limitations |
| FORTRAN Chain Card | *bbbbbCHAIN(21,2) | Starter Link Chain Card |
| FORTRAN Chain Link | Binary Cards | Starter Link |
| FORTRAN Chain Card<br>FORTRAN Chain Link | *bbbbbCHAIN(N,2)<br>Binary Cards | Nth Link to be Tested<br>Link N |
| FORTRAN Chain Card<br>FORTRAN Chain Link | Repeat of above for each link to be tested | |
| FORTRAN Data Card | *bbbbbDATA | Indicates Input Data |
| ABA and ABB Input | BCD and Octal Cards. Chain words for test links modified by adding 004000000000 | ABA and ABB Tables |
| Pseudo Instructions | BCD and Integers | Define Pseudo Program |
| Data | Floating Decimal, Integers, BCD (see Link Writeups) | Define Problem Parameters |

[1] Analyst's name; [2] Job number; [3] Number of lines of printed output; [4] Run time; [5] SAMIS physical tape number

# SECTION 3

# MASTER INTELLIGENCE

Master intelligence includes an object program and auxiliary subroutines. Its functions include positioning tapes, assigning storage, transferring library subroutines, handling the interpretation of the pseudo instructions and providing the basic error and logic routines of the program. Using open subroutines, MINTS directs the execution of operations required by the instructions FILL, SAVE, ERRS, SKIP, PAWS, STOP, and HALT. For all other instructions, MINTS selects a link to perform the required operation.

MINTS optionally provides a printed record of calculation progress. This can include statements of core data allocation before and after execution of each link and time just before entering a link.

On the following pages appears a detailed description of components of the MINTS link. Because of their importance, detailed descriptions of four of the subroutines are also included. The two auxiliary subroutines TAPES and COINS are used in every link.

Master Link: MINTS, Master Intelligence

Purpose: To make accessible required data, to select required operation subroutines, and to interpret and execute logic pseudo instructions.

Restrictions:

(1) Instructions must be included in the SAMIS pseudo instruction table (ABA or ABB) which has been stored in core before entry to MINTS.

(2) Data handling limitations are those imposed by the operation subroutines and operation links used.

Method: Each pseudo instruction is read and executed in turn until the program is completed.

There are two types of pseudo instructions --- logic instructions and operation instructions. The flow control instructions include ERRS, SKIP, STOP, PAWS and HALT. These are treated directly by MINTS. They cause the following actions:

(1) ERRS: If the previous link resulted in a recoverable error, this code directs continuation of calculations in the recoverable error mode. All instructions between the ERRS and succeeding SKIP or STOP will be executed in sequence. If a flow disrupt error has not occurred, all these instructions are disregarded.

(2) SKIP: If activated, the number of pseudo instructions denoted in the last field are skipped over and those following are executed. The instruction is activated if it is not paired with a previous ERRS instruction or in the recoverable error mode.

(3) STOP: If in the recoverable error mode, all succeeding instructions are ignored. If not in the recoverable error mode, this instruction is ignored.

(4) PAWS: The calculations are stopped. They can be restarted by pressing the start button. (This operation is not admissible in the IBM 7094-7040 D.C.S.)

(5) HALT: The CALL EXIT instruction is executed to terminate calculations.

MINTS actions under an operation instruction are locating data, calling the operation subroutine, and providing error control. Data is handled in accordance with the link designation numbers (See Section 4.2) as follows:

(1) Any matrix in core but not permitted in core is erased.

(2) All required matrices in core are retained. They are moved to the upper part of core.

(3) All matrices (referenced in the TRA block) not required but available and permitted in core are moved to the bottom of core.

(4) Any matrices required in core but not in core are read in from tape behind those required and already in core.

(5) Tapes are positioned for all output matrices and input matrices not in or required in core.

Two of the operation instructions are performed directly by MINTS. These are the instructions FILL and SAVE. These instructions cause the following actions:

(1) FILL: Matrices in the A and/or B and/or C fields are loaded into core. If a matrix is not indicated to be read in and a corresponding matrix is already in core, this matrix is retained, e.g., if the B fields are left blank, the B matrix already in core will be left intact.

(2) SAVE: Matrices A and/or B and/or C are moved from core to tape.

The operation link is called by CALL CHAIN (I,4) where I is the CHAIN number (Table 2-5). After execution control is returned to MINTS. If necessary, an error printout is made using ERROR.

Printout of the TRA block before and/or after each link provides a record of core allocation. Printout of computer clock time before link execution provides data and link execution times. These printouts are controlled by data in the ABB array read by MAKER.

Usage:

(1) Calling sequence is CALL CHAIN (26, 4)

(2) ABB(2) controls core allocation and time printouts as follows:

ABB(2) $<$ 0, printout time before operation link execution

ABB(2) = 0, do not printout core status or time

ABB(2) $>$ 0, printout core status before and after link execution and time before execution

Coding Information:

(1) This link includes a driving program and several special subroutines as follows:

| Label | Name | Purpose |
|---|---|---|
| Z1 | Subroutine ERROR | Printout status when errors occur |
| Z2 | Subroutine IN | Move data into core |
| Z3 | Subroutine SHIFT | Rearrange data in core |
| Z4 | Subroutine FINDS | Position tape |
| Z5 | Master Intelligence | Main Program |
| Z6 | Subroutine TSNAM | Test matrix name |

Subroutine with arguments in their calling sequences are called as follows:

CALL IN (LOCA, LOCB, LOCC, LE)

LOCA, LOCB, and LOCC are tape group numbers of matrices to be read in.
LE = initial index in the data region, to which data can be moved.

CALL SHIFT (IND, LZ)

IND = 1 if shift of data is required.
= -1 if shift is not required but is acceptable.
LZ = initial index in the data region to which data may be shifted.

CALL FINDS (LOC, INE)

INE indicates whether A1, B2, or C3 is of interest. If INE matches NTRA(10), A1 is of interest; if NTRA(20), B2; and if NTRA(30), C3.

(2) This link requires additional SAMIS subroutines as follows:

| Auxiliary Subroutines | | Other Subroutines | |
|---|---|---|---|
| Z 7 | TAPES | M 4 | BADIBM |
| Z 8 | COINS | | |

(3) This link requires a CLOCK subroutine whose function is to interrogate the computer clock and return the time. This subroutine is normally part of the system software at an installation.

(4) Use of the CHAIN instruction and instructions indicated by comment cards may not be admissible in all compilers.

MINTS Subroutine: FINDS, Tape Search

Purpose: To position a tape for reading or writing by an Operation Link.

Restrictions:

(1) Tape group numbers must be in sequence and increase as the tape is moved in the forward direction.

(2) The block number of the last block in each group must be negative and the same as the number of blocks. Block numbers of all other blocks in the group must be positive and in sequence starting from one.

(3) Data must be written on tape in the form written by TAPES.

Method: The tape to be searched is backspaced one block and the block read. If the tape group number is less than that desired, the tape is moved forward until it is positioned just before the group required. Completion of the reading of a group is determined by reading a negative block number. If the tape group number is greater than that desired, the tape is backspaced over the groups and positioned in front of the group required. Backspacing is controlled by the block number in the last block of each group. After backspacing over a group, the last block in the preceding group is read to determine how many blocks exist in the next group.

Usage:

(1) Calling sequence is CALL FIND (L,I) L indicates the tape group number being searched. It is a negative five digit number. The upper two digits are the logical tape number. I is the matrix referred to. If I = IA the A matrix tape is being positioned; if I = IB, the B matrix tape, and if I = IC, the C matrix tape.

(2) This subroutine uses TAPES as a subprogram.

Coding Information:

(1) Tape search can be started from any tape position between the beginning of the tape and the end of record gap immediately following the last block of the last group.

(2) Tape rewind is used by the subprogram only if positioning is to be before the first group.

(3) If the last block of a group does not contain a negative block number, the non-corrective error exit is taken.

MINTS Subroutine: ERROR, Diagnostic Printout

Purpose: To identify the type of error and state of the calculations when an error is encountered.

Restrictions: NONE

Method: A test of the error indicator (NCE) is made: if positive, a non-corrective error is printed; if negative, a recoverable error is noted. In any case, the pseudo instruction being executed when the error occurred is printed. In addition, identification information for the A, B, and C matrices is printed if available in core.

Usage:

(1) Calling sequence is CALL ERROR

(2) Direct transfer to this subroutine can be used at any time to print out the operative pseudo instructions and the matrix identification information. For this purpose instructions are included using ERROR as a subroutine.

Coding Information:

(1) There are no error stops in this subroutine.

Auxiliary Subroutine: TAPES, Tape read-write

Purpose: To handle all block reading and writing on matrix and scratch tapes, tape backspacing, and rewinding.

Restrictions:

(1) All blocks on tape must be at least 127 words long. The first 120 words are data, the last 7 words contain data identification.

(2) Tape numbers used must conform with SAMIS and local tape assignments.

Method: Blocks may be handled in groups or singly. Under group operation, blocks are read into or written from consecutive core locations. Identification information is simultaneously changed in the matrix identification region in core (TRA) as the blocks of data are read or written. In writing, all blocks are written and the last given a negative block number. In reading, as many blocks are read as in the group or the number specified, whichever is least. In individual operation, each block is written or read by separate subroutine entries. Block numbers are increased from the initial value in the TRA block and the final one is set negative.

Usage:

(1) Calling sequence is CALL TAPES (NT, NB, IL, A, J)

NT is the logical tape-group number $1000 < NT \leq 99999$. (NT = (Tape No.x 1000) + group number). Only the thousand digits are used as tape designations. If $NT < 0$, tape is moved backwards; if $NT > 0$, forwards.

NB is the maximum number of blocks to be handled. If $NB < 0$, blocks are to be read. If $NB > 0$, blocks are written. If NB = 0 rewind is performed. If $NT < 0$ and $NB > 0$, tape is backspaced NB blocks. $NB \leq 32767$ for a group.

IL is the index value of the first word of identification information for the matrix being handled. If $IL < 0$, blocks are being handled individually, if $IL > 0$ blocks are being handled in groups.

A is the array to be written or read.

J is the index of the first element in the array to be written or read.

(2) This subroutine requires COINS as a subroutine.

(3) Data and identification information must be stored in consecutive locations starting at the initial location. 120 words are taken or placed in core for each data block. The seven words of identification on tape correspond to the first eight matrix identification words in the TRA region.

Coding Information:

(1) After reading a block, the block sequence number of the last block read is left in the TRA region. After writing a block, the block number in the TRA region is increased by one unless the last block of a group has been written. In that case the block number is left unchanged.

(2) There are no non-corrective or flow disrupt stops in this subroutine.

(3) This program can be modified to convert from tape to disk storage any matrix (non-scratch) tape.

Auxiliary Subroutine: COINS, Code Interpreter and Coiner

Purpose: To form or disassemble a packed word to or from integer words defining row and column number.

Restrictions:

(1) The maximum number of gridpoint components must be less than 10, and may be zero.

(2) All packed words must be positive or zero; gridpoint numbers may be positive or negative but must be less than 4096 in absolute value.

Method: In coining from four words, sixteen is added to the component number and the sign of the node number changed if it is negative. Packing and unpacking is done by adding and shifting. 12 bits are reserved for node number and five for component number. These 17 bits are treated together when two word packing or unpacking is performed.

Usage:

(1) Calling sequence is CALL COINS (KOD, I)

KOD is the first location containing input data (code numbers)
I is the mode index and is interpreted as follows:
- -1, coin a one word code from 4 words of input
- 0, coin a one word code from 2 words of input
- 1, decompose one word into 2 words; a row and a column code
- -2, decompose one word into 4 words; a row gridpoint number, a row component number, a column gridpoint number, and a column component number.

(2) Input words and output words must be stored in consecutive locations starting at KOD(1). A maximum of four words are required.

Coding Information:

(1) This is a FAP subroutine.

(2) This program must be changed if the 36 bit word size is changed.

(3) The program has 96 instructions. If the four word coining and interpretation is separated, there will remain but 46 instructions.

(4) Normal operation will require an average of 32 machine cycles from entry to the linkage until return to the object program.

(5) No storage spaces are required other than those occupied by input, output, and the program.

(6) If a gridpoint number is greater than 4095 in absolute value, KOD(5) is set equal to one before return and the offending code put in KOD(1). If both node numbers are minus zero, the code is taken as zero.

# SECTION 4

# CALCULATION CONTROLS

The sequence of operations is controlled by the sequence of pseudo instructions entered as input by the analyst. In operation, the pseudo instructions are read by the Pseudo program generator (MAKER) and a special problem pseudo program written on tape for use during calculations. This section includes a description of the pseudo instructions and their handling by MAKER.

## 4.1 PSEUDO INSTRUCTIONS

The function of the pseudo instructions is to provide the user with a simple control by which he can direct calculations. The burden imposed on the analyst by this device is more than repaid by the flexibility of computer operation. The analyst can easily direct any sequence of operations he requires. Thus he can incorporate equilibrium checks and calculation of reactions. Reorganization of the data is at his disposal.

There are two types of pseudo instructions: logic instructions and operation instructions. The logic instructions direct writing of the problem pseudo program and provide pseudo program logic. The operation instructions call for links which operate on data arrays.

The format for both types of instruction is the same and is shown in Figure 4-1. Data is located in nine fields. The "O" field defines the instruction sequence number. Data in the A, B, C, and E fields is in fixed decimal form. The A, B, and C fields contain the tape storage assignment (tape-group number) for the A1, B2, and C3 matrices while field E contains control information for the operation to be performed. Fields 1, 2, and 3 contain the 6 character alphanumeric names of the A1, B2, and C3 matrices respectively. The last three characters must be numeric. The code field contains the four letter mnemonic code which specifies the subroutine or subprogram to be used.

| Field 0 | A | 1 | B | 2 | Code | C | 3 | E |
|---|---|---|---|---|---|---|---|---|
| F8.1 | 3X,I5 | 2X, A3,I3 | 3X,I5 | 2X, A3,I3 | 4X,A4 | 3X,I5 | 2X, A3,I3 | 2X,I6 |

PSEUDO INSTRUCTION FORMAT

Figure 4-1

## 4.2 LOGIC INSTRUCTIONS

The pseudo instruction shown in Figure 4-2 is a logic instruction. This card is required to cause a transfer of control to the success exit.

| Field 0 | A | 1 | B | 2 | Code | C | 3 | E |
|---|---|---|---|---|---|---|---|---|
| 14 | | | | | HALT | | | |

HALT PSEUDO INSTRUCTION

Figure 4-2

A list of the logic pseudo instructions and the program interpretation is given in Table 4-1. The PREP, VARY and BACK instructions provide the capability to write a loop in the pseudo instructions involving

TABLE 4-1

LOGIC PSEUDO INSTRUCTIONS

| Instructions | Interpretation |
|---|---|
| PREP | Prepare for multiple execution of the following instructions. Execute the instructions between the PREP and the next BACK instruction the number of times specified in the E field. |
| VARY | Vary matrix or tape numbers in the next instruction by augmenting corresponding field data by the specified integers after one pass. |
| BACK | Back up and repeat instructions after PREP. |
| ERRS | Disrupt errors can be corrected as follows |
| SKIP | Skip the next "n" pseudo instructions where n is specified in the E field. (Skip cannot be included between a PREP and BACK instruction). |
| STOP | Stop this case and go to HALT. |
| PAWS | Pause in the calculations. Operator can restart at any time. (Not operative on IBM 7094-7040 DCS). |
| HALT | Halt and indicate a successful exit. |

address modifications. Operation of a single instruction a multiple number of times with address modifications is provided by "serial" operation of a link and is defined by E field input. Detailed descriptions of serial capabilities and usage are included in link descriptions. Links capable of serial operation include ADDS, BILD, COLS, DECO, INKS, MULT, READ, ROWS, SORTS and SUBS.

The ERRS, SKIP and STOP instructions provide for including alternate paths in calculation flow when a recoverable error is encountered.

## 4.3 OPERATION INSTRUCTIONS

The pseudo instruction shown in Figure 4-3 is an operation instruction. This instruction requires that the A1 matrix, called MAT001 and stored as group 1 on tape 10, be added to the B2 matrix, called MBT002 and stored as group 1 on tape 9. The result is the C3 matrix, called MCT001 and is to be stored as group 1 on tape 12.

| 0 | A | 1 | B | 2 | Code | C | 3 | E |
|---|---|---|---|---|---|---|---|---|
| 3.5 | 10001 | MAT001 | 9001 | MBT002 | ADDS | 12001 | MCT001 | |
| 1 | 2 | 3 4 | 5 | 6 7 | 8 | 9 | 10 11 | 12 |

ADDS PSEUDO INSTRUCTION

Figure 4-3

It is seen that each matrix is specified by two fields of data. The first field defines the tape and tape group number. If this first word is left blank, the matrix is to be found or left in core. The second field contains letters and numbers which identify the matrix. The letters can be selected by the user. The numbers can be selected by the user, except that stiffness, stress, loading, and mass matrix numbers are chosen by the program to coincide with the element numbers.

An operation instruction can involve one or two cards. The second card, called a continuation card, has the same form as the first except that the operation code is CONT (continue). Only one continuation card is permitted with each operation instruction. Continuation cards cannot be used with FILL or SAVE operations.

The purpose of continuation cards is to provide the ability to address as many as six sets of matrices for a link. This capability is used only for the BILD operation. In this case, the A1 matrix on the continuation card defines the mass matrix.

To clarify documentation, matrices on the continuation card are designated as the F1, G2, and H3 matrices. These appear on the card in positions corresponding to A1, B2, and C3.

Instructions are performed in the sequence in which they are introduced. Since the instructions do not define the matrix size, they can be used for any problem requiring the same operational sequence.

A list of the operation pseudo instructions and their interpretation is given in Table 4-2. These operations provide for all aspects of matrix manipulation. Matrix generation is provided by BILD. Matrix arithmetic is provided by the ADDS, SUBS, MULT, CHOL, ITER, CHIN, and ROOT operations. Matrix element reordering, selecting, scaling, and reformating is provided by the SORT, ROWS, COLS, FLIP, WASH, CODE, and DECO operations.

TABLE 4-2

OPERATION PSEUDO INSTRUCTIONS

| Instructions | Interpretation |
|---|---|
| ADDS | Form C3 = A1 + B2 |
| BILD | Construct small deflection stiffness, stress, loading, and/or mass matrices as A1, B1, C1, and F1, respectively. |
| CHIN | Form B2 such that B2 $B2^T$ = A1 and C3 = $B2^{-1}$ where A1 is symmetric and positive definite. |
| CHCL | Form C3 = $A1^{-1}$ B2 using Choleski decomposition |
| CONT | Continuation Card (See pg 43) |
| CODE | Transform A1 to coded format as C3 |
| COLS | Put the A1 matrix in column listing and call it C3 |
| DECO | Transform A1 to precoded format as C3 |
| FILL | Read A1 B2, and/or C3 into core |
| FLIP | Form C3 = $A1^T$ |
| INKS | Print matrices A1, B2, and/or C3 |
| ITER | Form C3 = $A1^{-1}$ B2 using accelerated Seidel iteration. A1 must be positive definite. |
| MULT | Form C3 = A1 B2 |
| READ | Read matrices A1, B2, and/or C3 from cards. |
| ROWS | Put the A1 matrix in row listing and call it C3 |
| ROOT | Find latent roots and vectors of A1, a symmetric matrix. Let vectors be B2, roots C3. |
| SAVE | Write A1, B2, and/or C3 on tape |
| SORT | Sort a matrix A1 by row or column as C3 |
| SUBS | Form C3 = A1 - B2 |
| WASH | WASH A1 elements from B2 to produce C3 |

Input and output operations include READ, FILL, SAVE and INKS. Further details on the programs evoked may be found in Section 5.

Input and output of each link is summarized in Table 4-3. This table defines the inputs, input formats, input listing and output produced, output format, and output listing. These data are produced for each of the operating modes of the operation links. It is seen that most links put output in the same format and listing as input.

As a minimum, each link can treat matrix data when each of the matrices is assigned to a separate tape. In addition, most links have the capability to accept matrices already in core or to permit two or three matrix tape assignments involve the same tape.

Table 4-4 summarizes the acceptable matrix data assignments for each link. In this table, X, Y, and Z designate three distinct matrix tape numbers These numbers may be any of the available tapes for matrix or scratch storage (see Table 2-3) unless the scratch tapes are tied up by the link. When two or more symbols (e.g., X X X) are used in the table, the same tape is indicated for two or more inputs.

TABLE 4-3

OPERATION INPUT AND OUTPUT

| Link | Mode | Input Arrays | Format | Listing | Output Arrays | Format | Listing |
|---|---|---|---|---|---|---|---|
| ADDS | 1 | A1, B2 | Coded | Row | C3 | Coded | Row |
| | 2 | A1, B2 | Coded | Column | C3 | Coded | Column |
| BILD | 1 | Material Tables and Element Data | | | A1 | Coded | Row |
| | | | | | B2 | Coded | Row |
| | | | | | C3 | Coded | Column |
| | | | | | F1 | Coded | Row |
| CHIN | 1 | A1 | Coded | Row | B2 | Coded | Row |
| | | | | | C3 | Coded | Row |
| CHOL | 1 | A1 | Coded | Row | C3 | Coded | Column |
| | | B2 | Coded | Column | | | |
| CODE | 1 | A1 | Precoded | Row | C3 | Coded | Row |
| | 2 | A1 | Precoded | Column | C3 | Coded | Column |
| | 3 | A1 | Coded | Row | C3 | Coded | Row |
| | 4 | A1 | Coded | Column | C3 | Coded | Column |
| COLS | 1 | A1 | Coded | Row | C3 | Coded | Column |
| | 2 | A1 | Coded | Column | C3 | Coded | Column |
| | 3 | A1 | Coded | Unsorted | C3 | Coded | Column |
| DECO | 1 | A1 | Precoded | Row | C3 | Precoded | Row |
| | 2 | A1 | Precoded | Column | C3 | Precoded | Column |
| | 3 | A1 | Coded | Row | C3 | Precoded | Row |
| | 4 | A1 | Coded | Column | C3 | Precoded | Column |
| FLIP | 1 | A1 | Coded | Row | C3 | Coded | Column |
| | 2 | A1 | Coded | Column | C3 | Coded | Row |
| | | A1 | Coded | Unsorted | C3 | Coded | Unsorted |
| INKS | 1 | A1 | Coded or Precoded, Row or Column Listed | | | | |
| | | B2 | Coded or Precoded, Row or Column Listed | | | | |
| | | C3 | Coded or Precoded, Row or Column Listed | | | | |
| ITER | 1 | A1 | Coded | Row | C3 | Coded | Column |
| | | B2 | Coded | Column | | | |
| MULT | 1 | A1 | Coded | Row | C3 | Coded | Row or Col (See Link Writeup) |
| | | B2 | Coded | Column | | | |
| | 2 | A1, B2 | Precoded | Row or Col | C3 | Precoded | Row |
| READ | 1 | A1 | Coded, Row, Col, or Unsorted; Precoded, Row or Col | | | | |
| | | B2 | Coded, Row, Col, or Unsorted; Precoded, Row or Col | | | | |
| | | C3 | Coded, Row, Col, or Unsorted; Precoded, Row or Col | | | | |

| Link | Mode | Input Arrays | Format | Listing | Output Arrays | Format | Listing |
|---|---|---|---|---|---|---|---|
| ROOT | 1 | A1 | Precoded | Row | B2 | Precoded | Column |
| | | | | | C3 | Coded | Column |
| ROWS | 1 | A1 | Coded | Row | C3 | Coded | Row |
| | 2 | A1 | Coded | Column | C3 | Coded | Row |
| | 3 | A1 | Coded | Unsorted | C3 | Coded | Row |
| SORT | 1 | A1 | Coded | Row, Col, or Unsorted | C3 | Coded | Row or Col |
| SUBS | 1 | A1, B2 | Coded | Row | C3 | Coded | Row |
| | 2 | A1, B2 | Coded | Column | C3 | Coded | Column |
| WASH | | A1 | Coded | Row | | | |
| | 1 | B2 | Coded | Row | C3 | Coded | Row |
| | 2 | B2 | Coded | Column | C3 | Coded | Column |
| | 3 | B2 | Coded | Unsorted | C3 | Coded | Unsorted |

TABLE 4-4

OPERATION MATRIX TAPE ASSIGNMENTS

| Operation | Mode | A1 Tape | B2 Tape | C3 Tape | Limitations+ | Scratch Tapes Used |
|---|---|---|---|---|---|---|
| ADDS | 1 | X | Y | Z | None | None |
| | 2 | 0 | X | Y | A1 fits* | |
| | 3 | X | 0 | Y | B2 fits* | |
| | 4 | X | Y | 0 | C3 fits* | |
| | 5 | 0 | X | 0 | A1 and C3 fit* separately | |
| | 6 | X | 0 | 0 | B2 and C3 fit* separately | |
| | 7 | X | X | Y | A1 fits* or B2 in core$ | |
| | 8 | X | Y | X | C3 fits*$ | |
| | 9 | X | Y | Y | C3 fits*$ | |
| | 10 | 0 | X | X | C3 fits*$ | |
| | 11 | X | 0 | X | C3 fits*$ | |
| | 12 | X | X | 0 | A1 fits* or B2 in core$ | |
| BILD | 1 | X | Y | Z | None | None |
| CHIN | 1 | X | Y | Z | None | None |
| | 2 | X | Y | X | None$ | None |
| | 3 | X | Y | Y | None$ | None |
| CHOL | 1 | X | Y | Z | None | Both |
| | 2 | X | Y | X | None$ | Both |
| | 3 | X | Y | Y | None$ | Both |
| CODE | 1 | X | - | Y | None | None |
| | 2 | X | - | X | A1 one block long$ | None |
| COLS | 1 | X | - | Y | None | First, if A1 does not fit |
| | 2 | 0 | - | X | A1 fits* | |
| | 3 | X | - | X | None$ | First, if A1 does not fit |
| | 4 | 0 | - | 0 | A1 fits* | |
| | 5 | X | - | 0 | A1 fits* | |
| DECO | Same as CODE, See CODE | | | | | |

| Operation | Mode | A1 Tape | B2 Tape | C3 Tape | Limitations+ | Scratch Tapes Used |
|---|---|---|---|---|---|---|
| FLIP | 1 | X | - | Y | None | None |
| | 2 | 0 | - | 0 | A1 fits* | None |
| | 3 | 0 | - | X | A1 fits* | None |
| | 4 | X | - | X | A1 one block long | None |
| | 5 | X | - | 0 | A1 fits* | |
| INKS | Any Tape or Core Disposition | | | | None | None |
| ITER | 1 | X | Y | Z | None | Both |
| | 2 | X | Y | X | None$ | Both |
| | 3 | X | Y | Y | None$ | Both |
| MULT[1] | 1 | X | Y | Z | A1 or B2 fit* | First, if A1 and B2 do not fit* |
| | 2 | 0 | X | Y | A1 fits | |
| | 3 | X | 0 | Y | B2 fits* | |
| | 4 | 0 | 0 | X | A1 and B2 fit* | |
| | 5 | X | X | Y | A1 fits$ | |
| | 6 | X | Y | Y | A1 or B2 fit*$ | |
| | 7 | 0 | X | X | A1 fits*$ | |
| | 8 | X | 0 | X | B2 fits*$ | |
| | 9 | X | X | X | A1 fits*$ | |
| READ | Any Tape or Core Disposition | | | | None | None |
| ROOT | 1 | X | Y | Z | | None |
| | 2 | X | X | Y | None$ | None |
| | 3 | X | Y | Y | None$ | None |
| | 4 | X | Y | X | None$ | None |
| | 5 | X | X | X | None$ | None |
| ROWS | Same as COLS, See COLS | | | | | |
| SORT | Same as COLS, See COLS | | | | | |
| SUBS | Same as ADDS, See ADDS | | | | | |
| WASH | 1 | X | Y | Z | A1 fits* | None |
| | 2 | 0 | X | Y | A1 fits* | None |
| | 3 | X | 0 | Y | A1 and B2 fit* | None |
| | 4 | X | Y | 0 | A1 and C3 fit* | None |
| | 5 | 0 | 0 | X | A1 and B2 fit* | None |
| | 6 | 0 | X | 0 | A1 and C3 fit* | None |
| | 7 | X | 0 | 0 | A1 and B2 fit* | None |
| | 8 | 0 | 0 | 0 | A1 and B2 fit* | None |
| | 9 | X | X | Y | A1 fits*$ | None |
| | 10 | X | Y | X | A1 fits*$ | None |
| | 11 | X | Y | Y | A1 and B2 fit*$ | None |
| | 12 | 0 | X | X | A1 and B2 fit*$ | None |
| | 13 | X | 0 | X | A1 and B2 fit*$ | None |

[1] If precoded, A1 and B2 must fit in core.

| Operation | Mode | A1 Tape | B2 Tape | C3 Tape | Limitations+ | Scratch Tapes Used |
|---|---|---|---|---|---|---|
| WASH, | 14 | X | X | 0 | A1 and B2 fit*$ | None |
| Cont'd | 15 | X | X | X | A1 and B2 fit*$ | None |

* Fit is required in the data space in core. This involves 20,000 locations as set by MAKER.

$ When the same tape is used in two or more fields, group assignments must increase sequentially reading from A1 to C3.

+ Limitations are in addition to those described in the link writeups.

Whenever a pair of matrices are on the same tape, the second one listed must have a tape group number just 1 greater than the first. For example, tape group numbers for mode 10 of ADDS could be 10124, 09003, 10125. "0" indicates that the matrix is consigned to core storage or comes from there. It is noted that if data has been left in core from the previous operation, it will be taken from there rather than from tape even if a tape number is given.

## 4.4 COMMENT CARDS

Comment cards may be included in a pseudo program. These cards are merely printed out with the program to facilitate program identification. This card is indicated by a "C" punch in column 1. Any legitimate key-punched characters may occur in columns 2 through 72. These cards must be placed in the pseudo program before the HALT instruction.

## 4.5 SAMPLE PSEUDO PROGRAM

Table 4-5 is an example of the pseudo instructions for a pseudo program. The following is an explanation of operations directed by the pseudo instructions.

<u>Instruction 1.0, READ</u>:

This instruction causes the input tape to be read and the information stored as designated on the card. In this case the matrices CAT001, CAT002, and CAT003 are stored on tapes nine, ten and eleven, each as group one.

TABLE 4-5

EXAMPLE PSEUDO INSTRUCTIONS

| 0 | A | 1 | B | 2 | C | D | 3 | E |
|---|---|---|---|---|---|---|---|---|
| 1.0 | 09001 | CAT001 | 10001 | CAT002 | READ | 11001 | CAT003 | |
| 3.0 | 11002 | CAT004 | 11003 | CAT005 | READ | 11004 | CAT006 | |
| 4.0 | 09001 | CAT001 | 10001 | CAT002 | ADDS | | SAT001 | |
| 4.1 | | | | | PREP | | | 4 |
| 4.2 | | 000001 | 00001 | 000001 | VARY | | 000001 | |
| 4.3 | | SAT001 | 11001 | CAT003 | ADDS | | SAT002 | |
| 4.4 | | | | | BACK | | | |
| 5.0 | | SAT005 | | | INKS | | | |
| 6.0 | | | | | HALT | | | |

Instruction 3.0, READ:

This instruction stores matrices CAT004, CAT005, and CAT006 on tape eleven as groups two, three and four respectively. Note that instruction numbers need not be in sequence.

Instruction 4.0, ADDS:

This instruction causes matrix CAT001, which is stored on tape nine as group 1, to be added to CAT002, which is stored on tape ten as group 1. The result is called SAT001 and is kept in core.

Instruction 4.1, PREP:

This instruction tells the pseudo program generator MAKER to set up the coding to perform all instruction between 4.1 (PREP) and 4.4 (BACK) four times.

Instruction 4.2, VARY:

This instruction tells the pseudo program generator MAKER to increment the parameters on card 4.3 by the amount indicated for each time after the first that 4.3 is to be executed.

Instruction 4.3, ADDS:

This instruction, on the first pass adds SAT001, which is in core, denoted by no tape-group number, to CAT003, which is stored on tape 11 as group 1, and calls the result SAT002. On the second pass, this instruction will have been modified (according to 4.2) to add SAT002, in core, to CAT004 from tape 11, group 2.

Instruction 4.4, BACK:

This instruction tells the pseudo program generator MAKER the end of the group of instructions to be repeated. Thus PREP and BACK define the beginning and end of a pseudo program loop. These correspond to the FORTRAN DO and CONTINUE statements.

Instruction 5.0, INKS:

This instruction causes SAT005, the final result of the indexing, to be printed on the output tape.

Instruction 6.0, HALT:

This instruction "wraps up the program" and returns control to the job system of the computer.

## 4.6 PSEUDO PROGRAM GENERATOR

MAKER, the pseudo program generator, is used at the beginning of a SAMIS run only. It first sets up core for subsequent links. This operation involves reading the ABA and ABB tables which define logic and link control information. In addition, MAKER defines the number of locations and blocks that can be put in the data region before generating the pseudo program for the problem under consideration.

Table 4-6 defines the card input and formats for tables ABA and ABB which define operation instruction data. ABA is a table of the logic pseudo instructions. Table ABB contains each interpretable instruction name followed by the corresponding link designation, if not MINTS. The designation is packed in the form of Figure 2-1. The Row-code gridpoint number is interpreted as the CHAIN number. The Row-code component number and column code gridpoint number define the matrices that must be in core and may be in-core before link execution. These are binary coded, i.e., 4 designates A, 2 designates B, and 1 the C matrices. The column code component number (1, 2, or 3) designates the number of output matrices.

In generating the problem pseudo program, MAKER reads pseudo instructions, checks their logical acceptability, and writes the problem program on tape 3. It eliminates PREP, VARY, and BACK instructions by generating the pseudo instructions for each pass through the loop. During execution, it prints out the input pseudo instructions. Upon completion, it can print out the problem pseudo program.

The following pages contain a detailed description of the MAKER link.

TABLE 4-6

INSTRUCTION INPUT TABLES

Card 1 FORMAT (12(A4,2X)) Label

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| PREP | ERRS | SKIP | STOP | VARY | BACK | HALT | NEXT | PAWS | TABA |

Cards 2-9 FORMAT(4(A4,012)) Label

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| PAWS | 000001000000[1] | ERRS | 000004000000[2] | SKIP | 000004000000[3] | STOP | 000004000000[4] | | TABB1 |
| NEXT | 000010000000[5] | HALT | 000020000000[6] | MOVE | | ADDS | 000400000301 | | TABB2 |
| SUBS | 000400000301 | MULT | 000440000301 | CHOL | 000640000001 | ITER | 000540000001 | | TABB3 |
| WASH | 000504000301 | ROOT | 000744000201 | CODE | 000300000001 | DECO | 000340000001 | | TABB4 |
| COLS | 000140000201 | ROWS | 001140000201 | SORT | 000140000201 | READ | 000040000003 | | TABB5 |
| INKS | 001100000340 | BILD | 001000000003 | FLIP | 000240000201 | CHIN | 000600000001 | | TABB6 |
| FILL | | SAVE | | | | | | | TABB7 |
| | | | | CONT | | | | | TABB8 |

1 - Print control word (See pg 31)

2 - SAMIS library tape number (See Section 2)

3 - SAMIS debug tape number (See Section 2)

4 - SAMIS MINTS tape number (See Section 2)

5 - First assigned scratch tape number (See Table 2-3)

6 - Second assigned scratch tape number (See Table 2-3)

Operation Link: MAKER, Pseudo program generator

Purpose: To generate and write a set of pseudo instructions on the program tape from a set of input instructions. This program eliminates the instructions PREP, VARY, and BACK by developing additional instructions and checks pseudo instruction logic. It prints out input pseudo instructions and optionally prints out the pseudo program.

Restrictions:

(1) Pseudo instructions must be in the proper format.

(2) The maximum number of instructions between a PREP and a BACK must be $\leq 2083$.

(3) The last pseudo instruction must be a HALT.

(4) The total number of pseudo instructions in the pseudo program must not exceed storage on one tape.

Method: Pseudo instructions are read from the input tape and rewritten on the pseudo program tape until a PREP is encountered. Instructions, after a PREP, are accumulated in core until a BACK is found. Within this range, pseudo instructions are generated for each time through the loop. Address modifications required by each VARY are made.After the loop, card reading and writing continues until the HALT is found. Logic checks include checking that each PREP is followed by a BACK before a HALT is encountered, that no PREP lies between an ERRS and its companion SKIP or STOP card, that two consecutive PREPS are separated by a BACK, that for each instruction there exists a subroutine, and that pseudo card sequence numbers never diminish. On finding errors in the pseudo instruction input, suitable off-line comments are made. If the only error is in the sequence numbers of the pseudo instructions, program progress is not halted.

Usage:

(1) Calling sequence is CALL MAKER.

(2) Upon calling, the input tape must be positioned just before the first pseudo instruction.

(3) The error exit is taken only if no satisfactory cases are included in the input.

(4) Pseudo instruction card format is F8.1,2((3X,I5), (2X,A3,I3)), (4X,A4), (3X,I5), (2X,A3,I3), (2X,I6). Data has the following interpretation:

| | | |
|---|---|---|
| (F8.1) | : | Card Sequence number |
| (3X,I5) | : | "A" matrix storage assignment |
| (2X,A3,I3) | : | A matrix identification |
| (3X,I5) | : | "B" matrix storage assignment |
| (2X,A3,I3) | : | B matrix identification |
| (4X,A4) | : | Pseudo code |
| (3X,I5) | : | "C" matrix storage assignment |
| (2X,A3,I3) | : | C matrix identification |
| (2X,I6) | : | Control information |

(5) Print out of the pseudo program will be activated if the control information in the HALT instruction is non-zero.

Coding Information:

(1) This link includes a driving program and several special subroutines as follows:

| Label | Name | Purpose |
|---|---|---|
| B 0 | DRIVER | Main Program |
| B 1 | Subroutine MAKER | Directs subprogram handling |
| B 2 | Subroutine RE | Reads pseudo instructions |
| B 3 | Subroutine PR | Prepare a loop |
| B 4 | Subroutine VA | Process a VARY instruction |
| B 5 | Subroutine WR | Write program tape instruction |
| B 6 | Subroutine BS | Backspace program tape |
| B 7 | Subroutine PT | Prints pseudo instructions |

Calling sequences have no arguments.

(2) This link is used only at the beginning of SAMIS execution.

(3) The pseudo program tape 3 is left in rewind status upon exit from MAKER.

# SECTION 5

## STRUCTURAL MODELING LINKS

Modeling links generate coefficients for matrices of the structural equations using material properties and structural element data. Links generate stiffness, stress, loads, and mass matrices for various element representations --- rod, torque tube, elementary beam, and shear beam, and the flat triangular surface element called Facet. Input data required for modeling link operation is described in detail in Section 7, and from a programmer's viewpoint in link writeups that follow.

Modeling links read the tables of material properties first and store these in core. Each group of cards defining the geometry, temperature, weight, and gridpoint data for an element is then read and matrices generated for the element and written on tape. Matrix numbers are selected to correspond with element numbers and tape numbers with pseudo instruction requirements. Optionally input data and output matrices may be printed out. Other element data groups are read and matrices generated until all have been treated.

Core allocation for modeling link execution is different than for operation link execution. Modeling links involve a larger number of instructions and treat small amounts of data at a time. Most of core is used by the instructions and a small amount required for data. A representative core allocation is provided by the BILD link under IBSYS. Then, IBSYS subroutines require approximately 6442 locations, the BILD instructions occupy 12631 locations and 5995 spaces are needed for data. If only a small core is available (less than 32K), modeling links require division into several links.

Documentation of modeling links is subdivided into subroutine groups because of the multiplicity of subroutines involved. The first group contains the link driver program, data reading, and output subroutines and is given the link identification. Other subroutine groups define the sets of subroutines for each type of structural element treated.

Writeups of modeling links that follow include tables 5-1 through 5-4 defining the matrix row and column coding generated for the structural matrices and its interpretation in terms of the structural problem.

Tables 5-1 and 5-2 apply to all types of elements. These tables define the row and column identification system used in the equations of motion. These tables apply to the stiffness, load, and mass matrices. Tables 5-3 and 5-4 define the row and column identification system for the stress-displacement equations. These tables apply to the stress matrix for the facet and line element.

Modeling Link: BILD, Constructor of small deflection stiffness, stress, and loading matrices (including the inertia matrix).

Purpose: To generate from material tables and element data and store on tape stiffness (A1 output) and/or stress (B2 output), loading (C3 output), and mass (F1 output) matrices.

Restrictions:

(1) The input tape must contain the material tables and element input data and be positioned to read the material tables.

(2) All output matrices must be written on tapes in coded format. A1, B2, C3, and F1 tape numbers must be different. If no tape assignment is given, matrices of that tape will not be generated. A1, B2, and F1 will be row listed. C3 will be column listed.

(3) Element data must correspond with that required by available subprograms (see group subprogram writeups).

Method: The material tables are read from the input tape. Element data for an element is read in, data checked, data optionally printed out, and transfer made to the required generation subroutine. This procedure is repeated for each element. Matrix numerical names are assigned to correspond with the element number. Tape assignments are consecutive starting from the initial assignment of the pseudo instruction.

Usage:

(1) Calling sequence is CALL CHAIN (16, 4).

(2) The materials table must be ended with a card of zeros. Data for a given material must be grouped together and ordered by increasing temperature. Material table data is generally in E8.4 format. Two cards are required for each temperature level. Information on these cards is given the following interpretation:

(2X,A6): The material identification
(E8.4): Temperature related to the given properties
(E8.4): Coefficient of thermal expansion, inches/inch degree Rankine
6(E8.4): D11, D21, D22, D31, D32, D33, material coefficients $lbs/in^2$
9(E8.4): D41, D42, D43, D44, D55, D65, D66, material coefficients $lbs/in^2$, (second card)
Any of the data may be zero except the material identification.

(3) Element input data must include at least two cards of data. Each card is numbered in the first column in Format I1 with a number from one to nine. The first card of data for an element must be numbered one and the last, nine. Other cards for the element need not be in sort. Omitted data is assumed to be zero. Format of all cards is (I1, 1X, I3, 1X, I2, 9E7.0, A2).

Interpretation of data is given in Section 7. Card data with the same interpretation by all BILD subroutine groups is summarized in the following table. Data interpreted differently by different groups are defined in group write-ups. Those included in the table are noted by (A).

| Card | Columns | Format | Data |
|---|---|---|---|
| 1-9 | 1 | I1 | Card number |
| 1-9 | 2-4 | I3 | Element number |
| 1-9 | 5-7 | IX,I2 | Element type (A) |
| 1 | 71-72 | A2 | Structural material identification |
| 5 | 8-70 | 9E7.0 | Local coordinate system location data |

(4) The absolute value of the pseudo instruction E field input/100 defines the number of elements to be handled. If the number is negative, the material tables and element input are written on the output tape. Matrices will be generated only if a tape assignment is given for them in the pseudo instruction. The last digit defines mass matrix assumptions. 0 and 5 denote finite difference; 1 and 6, potential energy; and 2 and 7, modified potential energy. If the last digit is $>4$, only the upper half of the stiffness matrix is produced.

(5) An error exit will occur if cards are in improper format or sort.

Coding Information:

(1) This link includes a driving program and special subroutines as follows:

| Label | Name | Purpose |
|---|---|---|
| PO | DRIVER | Main Program |
| PA | Subrcutine BILD | Read data, select subprogram group |

| Label | Name | Purpose |
|---|---|---|
| PH | SFT | Transforms stiffness loading and stress matrix to final coordinates, and eliminates negligible elements. |
| PP | TABLE | Interpolates in the material tables for element properties. |
| PQ | MOD | Modifies the transformation matrix for local coordinates. |
| PR | COES | Codes and writes matrices on tape. |
| PB,PC,PD, PE,PG,PI, PJ,PK,PL, PM,PN,PO | Subroutine Group FACET | Generate facet matrices. |
| PU,PV,PW, PX,PZ | Subroutine Group BEAM | Generate beam matrices. |

Arguments for calling sequences in the subroutine groups are defined in the group writeups. Other subroutines above with arguments are called as follows:

CALL TABLE (N1, A2, A3, A4, A5, A6, A7, A8, A9, A10, A11, A13, A14, A15, A16)

N1 is the material identification; A1, A2, .....A16 are the interpolated material characteristics.

CALL MOD (I, J, M, A)

A is the array to be modified; I, J are the row and column of A to initiate modification; M is the sequence number of the last row to be treated.

CALL COES (IM, A, N7)

IM is the location of the TRA block identifying the output; A is the array containing output; N7 is the matrix subscript number.

(2) This link requires additional SAMIS subroutines as follows:

| Auxiliary Subroutines | | Other Subroutines | |
|---|---|---|---|
| Z 7 | TAPES | M 4 | BADIBM |
| Z 8 | COINS | | |

(3) If new subroutine groups are to be added, additional branches must be provided as indicated in the comment cards of the source deck.

(4) Link designation is (16, 0, 0, 0).

BILD Subroutine Group: FACET, Facet element modeler

Purpose: To generate and store on tape the stiffness (A1 output), stress (B2 output), and/or loading (C3 output) and/or mass (F1 output) matrices in coded format for a flat triangular facet of uniform thickness.

Restrictions:

(1) Adequate element data must be introduced to define the FACET.

(2) Coordinate systems must be chosen to be consistent with the rest of the elements.

Method: The FACET package consists of 16 subroutines called, as required, by the FACET subroutine. These first generate the facet natural coordinates. In this coordinate system the loading matrix coefficients and the stiffness matrix coefficients are generated. The coefficients of the matrix to transform to the local coordinate system are then developed and the loading and stiffness matrices transformed to this system. The transformation coefficients relating the final and local system are generated. The loading matrix is then transformed to this system, coded, and written as the C3 matrix. The stiffness matrix is postmultiplied by the transformation matrix to put the displacements in the final coordinate system. If required, the transformation matrix coefficients which convert the stiffness into the stress matrix are then calculated and the transformation performed. The stress matrix is written on tape as the B2 matrix. The stiffness matrix is transformed to the final coordinate system and written on tape as the A1 matrix.

All matrices are identified by the alphabetic designation given on the BILD pseudo instruction but <u>are numbered to correspond with the related element number</u>. Matrices are stored in consecutive groups on the tapes designated on the BILD pseudo instruction. Equations used in the program are developed in the Structural Analysis and Matrix Interpretive System Technical Report.

Usage:

(1) Calling sequence is CALL FACET

(2) Element data required is contained on cards one, and nine.

The following information concerns the particular interpretation of element input data by the FACET subroutine. The sequence in which gridpoints are input defines the positive direction of the local normal. This is described in the Technical Report.

a. The "structural type" number is 31 (Card 1, field 1)

b. "Mass level": $\geqslant 0$ mass per unit area ((lb-sec$^2$)/in$^3$)

$< 0$ mass of Facet ((lb-sec$^2$)/in) (Card 1, field 9).

c. "Pressure level": Normal pressure in the local plus z direction in pounds per square inch (Card 2, field 6).

d. "Surface temperatures": The temperatures of the upper and lower surfaces in that order referenced to the zero (degrees Rankine) stress temperature state (Card 2, fields 8 and 9).

e. Thickness of Facet (Card 1, field 10). (in.)

f. The local coordinate system is a rectangular cartesian system. Material coefficients are assumed to be in the local coordinate system. (See Section 7.3).

g. Card 9 contains the gridpoint coordinates of the Facet (fields 2-10). Card 1 contains the corresponding elastic gridpoint numbers (fields 2-4) and substitute gridpoint numbers (fields 6-8). If substitute gridpoints are used, their coordinates are given on card 7, fields 2-10. (See Section 7.3). Cards 3, 4, 6, and 8 are ignored if included.

(3) A nonrecoverable error will occur if the element data does not define a Facet.

Coding Information:

(1) The purpose of each of the special subprograms is as follows:

| Label | Subroutine | Purpose |
|---|---|---|
| PB | FACET | Controls selection of other subroutines. |
| PC | COR | Finds natural coordinates and moments of inertia |
| PD | TRN | Forms transformation matrix from local to final coordinate systems |
| PE | LOD | Forms loading matrix coefficients in local coordinate system |
| PF | STF | Forms stiffness matrix in natural and local system |
| PG | STR | Forms matrix to transform gridpoint displacements to mean stress resultants |

| Label | Name | Purpose |
| --- | --- | --- |
| PI | STFQ | Dummy for buckling |
| PJ | TRS | Transposed a 3 x 3 matrix |
| PK | ADM | Relocates and adds 3 x 3's to form transformation matrices |
| PL | SAD | Forms a component of the stiffness matrix |
| PM | SCA | Multiplies a 3 x 3 matrix by a scalar |
| PN | ADD | Adds 3 x 3 matrices |
| PO | MUL | Multiplies 3 x 3 matrices |
| PT | SPR | Defines moment reaction for Facet normal forces. |

Calling sequences for subroutines with arguments in the calling statement are interpreted as follows:

CALL STFQ (N)
  N is the type of buckling matrix desired

CALL TRS (A, B)
  A is the 3 x 3 matrix transpose of B
  B is the 3 x 3 input matrix

CALL ADM (A, B, I, J)
  A is the 15 x 15 matrix to contain answer
  B is the 3 x 3 to be relocated
  I is the row partition number in A for answer
  J is the column partition number in A

CALL SAD (A, B, C, D)
  A is the answer array, 3 x 3
  B is an input array, 3 x 3
  C is a scalar
  D is the array to be multiplied by C and added to B, 3 x 3

CALL SCA (A, B, C)
  A is the answer array, 3 x 3
  B and C are input arrays, 3 x 3 to be added

CALL ADD (A, B, C)
  A is the matrix sum of B and C
  B is a 3 x 3 array
  C is a 3 x 3 array

CALL MUL (A, B, C)
A is the matrix product of B and C
B and C are 3 x 3 matrices

CALL SPR (A, AX, AY, X, Y, B)
A is the 3 x 3 matrix defining z forces for Facet
AX and AY are 3 x 3 matrices defining joint moments due to the z forces
X and Y are 3 x 1 vectors of Facet coordinates
B is a scalar (.5)

(2) This subprogram uses the DAT region for its erasable storage.

(3) Stiffness, loading, and mass matrix coding is as defined in Tables 5-1 and 5-2. Stress matrix coding is defined in Table 5-3.

TABLE 5-1

**Stiffness Matrix Row/Column Codes;**
**Stress Resultants Matrix Column* Codes; Loading Matrix Row Codes**

Table of Gridpoint - Component Numbers

| Row/Column | Gridpoint | Component | Symbol | Description |
|---|---|---|---|---|
| 1 | First Gridpoint Number | 1 | $u_{x_1}$ | x displacement; first gridpoint |
| 2 | " | 2 | $u_{y_1}$ | y displacement; first gridpoint |
| 3 | " | 3 | $u_{z_1}$ | z displacement; first gridpoint |
| 4 | " | 4 | $\phi_{x_1}$ | Rotation about x; first gridpoint |
| 5 | " | 5 | $\phi_{y_1}$ | Rotation about y; first gridpoint |
| 6** | " | 6 | $\phi_{z_1}$ | Rotation about z; first gridpoint |
| 7 | Second Gridpoint Number | 1 | $u_{x_2}$ | x displacement; second gridpoint |
| 8 | " | 2 | $u_{y_2}$ | y displacement; second gridpoint |
| 9 | " | 3 | $u_{z_2}$ | z displacement; second gridpoint |
| 10 | " | 4 | $\phi_{x_2}$ | Rotation about x; second gridpoint |
| 11 | " | 5 | $\phi_{y_2}$ | Rotation about y; second gridpoint |
| 12** | " | 6 | $\phi_{z_2}$ | Rotation about z; second gridpoint |
| 13 | Third Gridpoint Number | 1 | $u_{x_3}$ | x displacement; third gridpoint |
| 14 | " | 2 | $u_{y_3}$ | y displacement; third gridpoint |
| 15 | " | 3 | $u_{z_3}$ | z displacement; third gridpoint |
| 16 | " | 4 | $\phi_{x_3}$ | Rotation about x; third gridpoint |
| 17 | " | 5 | $\phi_{y_3}$ | Rotation about y; third gridpoint |
| 18** | " | 6 | $\phi_{z_3}$ | Rotation about z; third gridpoint |

* Stress matrix has an additional column (thermal stress) with a gridpoint number of zero and a component of 5.

** These rows are deleted if the gridpoint is considered in local coordinates. In this case all following row numbers are reduced by one. Dimensions for the stiffness matrix may vary from 15 x 15 to 18 x 18, for the stress matrix from 8 x 16 to 8 x 19, and for the loading matrix from 15 x 5 to 18 x 5.

TABLE 5-2

Loading Matrix Column Codes**

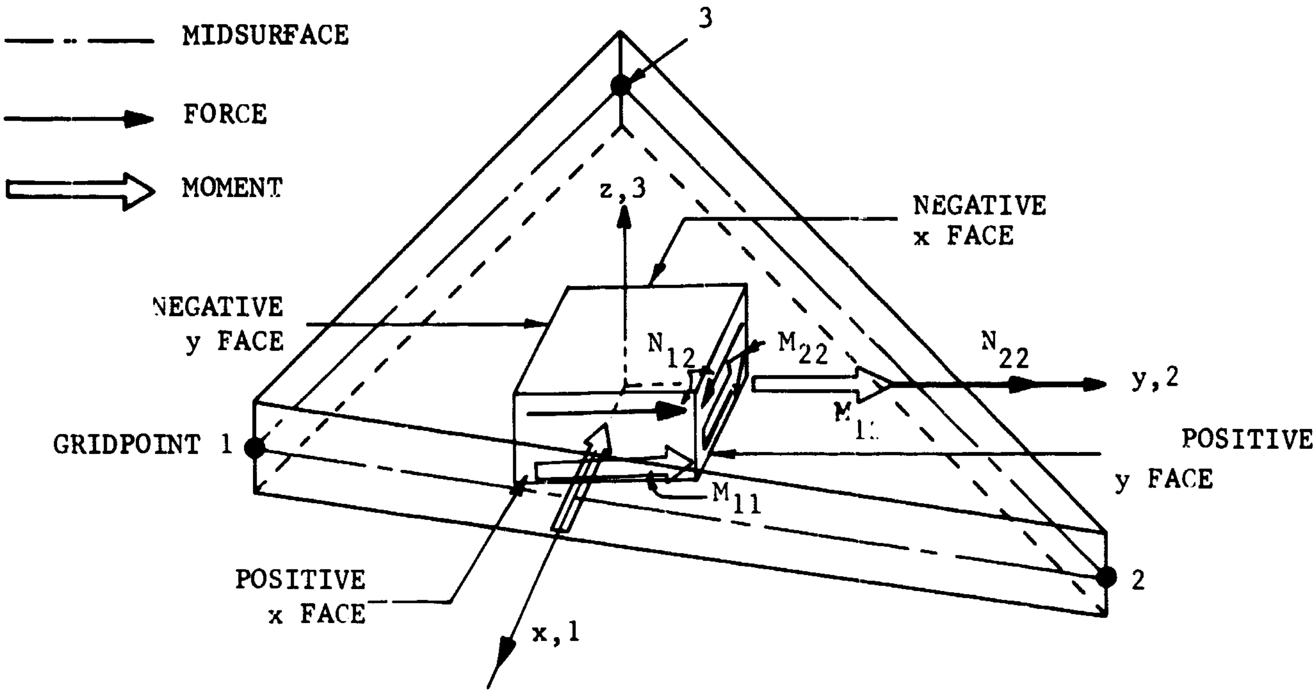
Table of Gridpoint - Component Numbers

| Column | Gridpoint | Component | Description |
|---|---|---|---|
| 1 | 0 | 1 | Gravity loading applied in overall x direction |
| 2 | 0 | 2 | Gravity loading applied in overall y direction |
| 3 | 0 | 3 | Gravity loading applied in overall z direction |
| 4 | 0 | 4 | Pressure loading in local z direction |
| 5 | 0 | 5 | Thermal loading |

** Loading row codes are the same as stiffness matrix row codes.

TABLE 5-3

STRESS-RESULTANTS MATRIX ROW CODES* (FACET)




| Row | Gridpoint | Component | Symbol | Description$ |
|---|---|---|---|---|
| 1 | Element No. | 1 | $N_{11}$ | Stress resultant in x-direction. Positive if causes tensile stresses on +x-face |
| 2 | Element No. | 2 | $N_{22}$ | Stress resultant in y-direction. Positive if causes tensile stresses on +y-face |
| 3 | Element No. | 3 | $N_{12}$ | Tangential shear. Positive if causes stress on +x-face (+y-face) in +y-direction (or +x-direction) |
| 4 | Element No. | 4 | $M_{11}$ | Stress couple in (xz)-plane. Positive if causes tensile stresses on +z-fibers in x-direction |
| 5 | Element No. | 5 | $M_{22}$ | Stress couple in (yz)-plane. Positive if causes tensile stresses on +z-fibers in y-direction |
| 6 | Element No. | 6 | $M_{12}$ | Twist moment. Positive if has the same effect as $N_{12}$ when z is positive |

$ $N_{11}$, $N_{22}$ and $N_{12}$ are in "force/length" units. $M_{11}$, $M_{22}$ and $M_{12}$ are in "moment/length" units. (xyz) refers to the local coordinate system.

* Stress column codes are the same as stiffness matrix column codes.

BILD Subroutine Group: BEAM, Beam element modeler

Purpose: To generate and store on tape the stiffness (A1 output) and/or stress (B2 output) and/or loading (C3 output) and/or mass (F1 output) matrices in coded format for a straight line element. The element may be a rod, tube, or beam or a composite of these.

Restrictions:

(1) Adequate data must be introduced to define a line element with uniform section properties.

(2) Coordinate systems must be chosen to be consistent with the rest of the elements.

Method: The BEAM package consists of 7 subroutines called, as required by the BEAM subroutine. These first generate the beam natural coordinates. In this coordinate system the loading matrix coefficients and the stiffness matrix coefficients are generated. The coefficients of the matrix to transform to the local coordinate system are then developed and the loading and stiffness matrices are transformed to this system. The transformation coefficients relating the final and local system are generated The loading matrix is then transformed to this system, coded, and written as the C3 matrix. The stiffness matrix is postmultiplied by the transformation matrix to put the displacements in the final coordinate system. If required, the transformation matrix coefficients which convert the stiffness into the stress matrix are then calculated and the transformation performed. The stress matrix is written on tape as the B2 matrix. The stiffness matrix is transformed to the final coordinate system and written on tape as the A1 matrix.

All matrices are identified by the alphabetic designation given on the BILD pseudo instruction but are numbered to correspond with the related element number. Matrices are stored in consecutive groups on the tapes designated on the BILD pseudo instruction. Equations used in the program are developed in the Structural Analysis and Matrix Interpretive System Technical Report.

Usage:

(1) Calling sequence is CALL BEAM

(2) Normally, element data required is contained on cards one, two, four and nine.

The following information concerns the particular interpretation of element data by the BEAM subroutine group. General requirements and definitions of input data are given in Section 7.

a. The "structural type" number is 21 or 22. 21 is the elementary beam and 22, the shear beam. (Card 1, field 1).

b. "Mass level": (Card 2, field 5) $>0$ mass per unit length ($lb\text{-}sec^2)/in^2$).

   $<0$ mass of the line element ($lb\text{-}sec^2)/in$).

c. "Pressure level": Normal pressure in the local plus z direction in pounds per inch. (Card 2, field 6).

d. "Surface Temperatures": The temperature change inducing elongation along the elastic line, $T_1$, and the temperature gradients inducing y and z axis curvatures; $T_2$ and $T_3$ respectively. (Card 2, fields 8, 9, and 10) (Degrees Rankine).

e. "Basic Geometric Data": $A_i$, $I_i$, i - 1, 2, 3 where $A_i$ are effective crossectional areas and $I_i$ effective moments of inertia. $A_1$ and $I_1$ resist force and a torque vector along the elastic line, the x axis; $A_2$, $I_2$, x-y shear and y moment of inertia respectively; $A_3$, $I_3$, x-z shear and z axis moment of inertia. The local y axis lies in the plane defined by the three gridpoints given. (Card 1, fields 5 through 10). (See Section 7.3).

f. The local coordinate system is a rectangular cartesian system. Material coefficients are assumed to be in the local coordinate system ( x lies along the line ). The use of the local coordinate system also simplifies description of gridpoint coordinates. (See Section 7.3).

g. Gridpoint information is given for the two ends of the line. A third gridpoint is used with the first two to define a principal plane of the member. For a symmetric beam, this may be either the plane of the web or a plane normal to the web through the neutral axis. In any event, the three points define the local x-y plane. Elastic gridpoint coordinates are given on card 9, fields 2-10. Corresponding gridpoint numbers are given on card 1, fields 2-4. (Note that the gridpoint number for the third gridpoint may be omitted). Substitute gridpoint coordinates are given on card 7, fields 2 and 3, and their coordinates on card 7, fields 5-10. Cards 3, 4, 6, and 8 are ignored, if included.

(3) A nonrecoverable error will occur if the element data is inadequate to define a line element.

Coding Information:

(1) The purpose of the special subroutines is as follows:

| Label | Subroutine | Purpose |
|---|---|---|
| PU | TN | Obtains line local coordinates, forms transformation matrix from local to final coordinate system |
| PV | LD | Forms loading vectors |
| PW | SF | Forms stiffness matrix in local coordinate system |
| PX | SR | Forms stress matrix |
| PZ | BEAM | Directs selection of subroutines |

None of these subroutines has calling arguments.

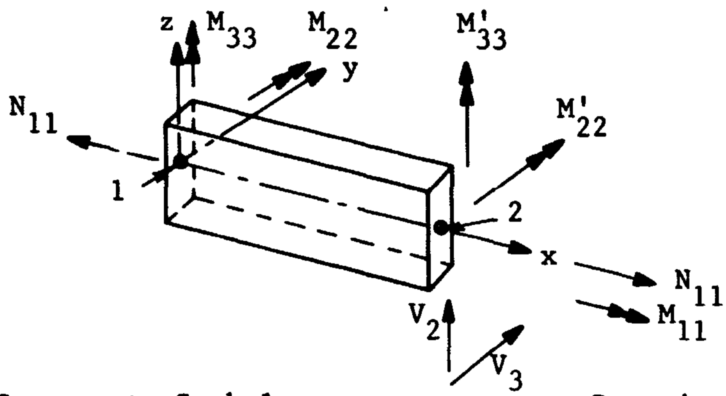
(2) This program uses the DAT region for its erasable storage.

(3) Stiffness, loading, and mass matrix coding is as defined in Tables 5-1 and 5-2. Stress matrix coding is defined in Table 5-4.

TABLE 5-4

Stress-Resultant Matrix Row Codes* (Line Element)

Table of Gridpoint - Component Numbers




| Row | Gridpoint | Component | Symbol | Description |
|---|---|---|---|---|
| 1 | END 1 | 1 | $N_{11}$ | Normal force in x direction, ends 1 and 2 |
| 2 | END 1 | 4 | $M_{11}$ | Twisting moment, x vector ends 1 and 2 |
| 3 | END 1 | 5 | $M_{22}$ | Bending moment y vector |
| 4 | END 1 | 7 | $V_2$ | Shear force in x-z plane |
| 5 | END 1 | 8 | $V_3$ | Shear force in x-y plane |
| 6 | END 1 | 9 | $M_{33}$ | Bending moment, z vector |
| 7 | END 2 | 5 | $M'_{22}$ | Bending moment, y vector |
| 8 | END 2 | 9 | $M'_{33}$ | Bending moment, z vector |

* Stress column codes are the same as stiffness matrix column codes.

# SECTION 6

# OPERATION LINKS

The operation links direct all calculations with problem-data. They may read matrix tape input and/or card input. They may produce printout and/or matrix tape output. Upon completion of execution of each link, control is returned to MINTS.

A special feature of some of the links is their ability to repeat operation without requiring a return to MINTS. This is called "serial" operation. To avoid tape searching, matrices are required to have sequential group numbers. Serial operation is usually indicated when the E field input is greater than 100.

Some of the links provide special data printouts. These are BILD, ITER, READ, and ROOT. These are activated by a negative number in the E field. BILD and READ provide printouts of input data. ITER provides printouts of iteration progress. ROOT provides a printout of structural frequencies calculated.

The pages following contain a detailed description of each of the operation links except the generation link BILD. This link was described in Section 5. The links in this section use the data region of core reserved by MAKER (20,000 locations) exclusively for matrix data. Thus this region contains only matrix codes and elements.

The theoretical basis for link calculations is described in the Structural Analysis and Matrix Interpretative System Technical Report. The writeups that follow describe the salient characteristics of the computational methods and programming details of the links.

Operation Link: ADDS, Matrix Adder

Purpose: To form C3 = A1 + B2 where A1, B2 and C3 are coded matrices, and to add series of matrices.

Restrictions:

(1) A1 and B2 must be in the same listing, either row or column. C3 will be in the same listing as A1 and B2.

(2) Either A1 or B2 may be in core or both may be on tape. If C3 does not fit in core, a tape assignment must be specified for it.

(3) For serial operation, either $C3 = \sum_{i=1}^{N} A1_i$ where the $A1_i$ are on tape and have like alphabetic names and numbers in sequence or $(C3_i = A1_i + B2_i)$ i = 1, N where $A1_i$, $B2_i$, and $C3_i$ are assigned to tapes. $A1_i$, $B2_i$, and $C3_i$ must be on different tapes, and be in sequential groups.

Method: If B2 is omitted or in core, the $A1_i$ are combined in core in a single pass. Elements with like codes are added. When the matrix would become too large for core, the block in core containing the lowest codes is written on the C3 tape to provide additional space. If codes less than those written on C3 reoccur, a second partition of C3 is formed, the new partition written on C3, the partitioning noted, and the recoverable error exit taken upon completion of calculations.

If both A1 and B2 are on tape, code-element pairs are selected in increasing code order from A1 and B2. Only one C3 matrix can occur for each A1 - B2 pair since ADDS can treat any size array in this mode of operation.

In serial operation, link instructions are recycled after changing control information.

Usage:

(1) Calling sequence is CALL CHAIN (8, 4)

(2) The information in the E field input defines 100N for serial addition. If N = 0, it is assumed to be 1.

(3) When C3 is formed of more than one partition, the recoverable error exit (NCE = -1) is taken.

Coding Information:

(1) This link includes a driving program and several special subroutines as follows:

| Label | Name | Purpose |
|---|---|---|
| H 0 | DRIVER | Main Program |
| H 1 | Subroutine ADDS | Directs Calculations |
| H 2 | Subroutine SWITCH | Transposes Codes |

Only SWITCH has calling arguments. It is called by CALL SWITCH (LA, NDIM).
LA is the array containing the codes to be transposed.
NDIM is the highest index of LA to be transposed.

(2) This link requires additional SAMIS subroutines:

| Auxiliary Subroutines | | Other Subroutines | |
|---|---|---|---|
| Z 7 | TAPES | N 2 | MIXES |
| Z 8 | COINS | N 3 | ORDER |
| | | N 4 | ORDERS |
| | | M 4 | BADIBM |

(3) Link designation is (8, 0, 6, 1).

Operation Link: CHIN, Choleski Decomposer and Inverter

Purpose: To form B2 = U and C3 = $U^{-1}$ where $U^TU$ = A1. U is an upper triangular matrix as is $U^{-1}$.

Restrictions:

(1) A1 must be on tape. It must be square, positive definite, and is assumed symmetric. Only the upper triangular matrix needs to be provided.

(2) The diagonal and upper triangular part of A1 must fit in core in variable bandwidth form with two vectors of a length equal to the order of the A1 matrix.

(3) B2 and C3 will be coded and listed by row. They must be assigned to tape. B2 will fit in core, but C3 may be larger than core.

Method: Coded elements of A1 are read and a table of distinct row codes compiled. The "A1" tape is backspaced and reread and the matrix put in variable bandwidth form, a list of the last element in each row being formed. A1 is then decomposed such that $U^TU$ = A1. If square roots of negative numbers are required, the matrix is not positive definite and the non-corrective error exit is taken. $U^{-1}$ is formed by column, defining C3, and the result coded and written on tape.

Usage:

(1) Calling sequence is CALL CHAIN (12, 4).

(2) Codes in B2 and C3 will correspond with those in A1.

Coding Information:

(1) This link includes a driving program and several special subroutines as follows:

| Label | Name | Purpose |
|---|---|---|
| L 0 | DRIVER | Main Program |
| L 1 | Subroutine CHIN | Directs Calculations |
| L 3 | Subroutine CHOL 2 | Decomposes matrix |
| L 5 | Subroutine INVS | Solves for inverse |
| L 6 | Subroutine CODTRI | Codes triangular matrix |
| L 7 | Subroutine POWER | Dummy routine |

Calling sequences with arguments are

CALL CHOL 2 (A, N, M 1)
A is the variable band matrix array
N is matrix order
M 1 is disregarded

CALL INVS (A, N, I 8)
A is the variable band matrix array
N is the matrix order
I 8 is the number of inverse matrix vectors required

CALL CODTRI (I C, KEY, KCO)
I C is the TRA index of output identification data block.
KEY is the first index of the key table in the DAT array.
KCO is the first index of the code table in the DAT array.

CALL POWER (A, N, D)
A is the variable band matrix array
N is the matrix order
D is the eigenvalue

(2) This link requires additional SAMIS subroutines as follows:

| Auxiliary Subroutines | Other Subroutines |
|---|---|
| Z 7 TAPES | M 2 CODTAB |
| Z 8 COINS | M 4 BADIBM |

(3) Maximum matrix size is 6666 x 6666. The actual limit depends on the bandwidth which is effected by the choice of gridpoint numbers in the problem. Expected capability is about 500th order.

(4) The non-corrective exit is taken if A1 is not positive-definite as well as for format, and listing errors.

(5) Link designation is (12, 0, 0, 2).

Revision 1
December 15, 1966

JPL Technical Memorandum No. 33-307

Operation Subprogram: CHOL, Simultaneous Equation Solver Using Choleski Triangular Decomposition

Purpose: To form $C3 = A1^{-1} B2$ where A1 is a square positive definite matrix and B2 a set of vectors.

Restrictions:

(1) A1 must be on tape. A1 must be square and positive definite, row listed and is assumed symmetric. Only the upper triangular part needs to be provided.

(2) B2 must be on tape and be listed by column and have a maximum of 500 columns. If B2 is omitted, the inverse of A1 is generated.

(3) C3 must be written on other than the B2 tape and will be coded and column listed. C3 may be larger than core.

(4) The diagonal and upper triangular part of A1 must fit in core in variable band form along with two full columns of the B2 matrix.

Method: Coded elements of A1 are read and a table of distinct codes compiled. The coded B2 matrix is read and written on scratch tape followed by the code table. The A1 matrix is then read into core and put in variable bandwidth form from a list of the locations of the last element in each row being formed. A1 is then decomposed such that $L L^T = A1$ where L is a lower triangular matrix. If square roots of negative numbers are required the matrix is not positive definite and the non-corrective error exit is taken. Otherwise, the equations Ly = B2 are solved, y replacing B2 in core. Then, lx = y is solved, x replacing y. Each column of B2 is treated separately and columns of C3 written on the scratch tape. Finally, the columns of C3 are read from the scratch tape, coded, and written on the output matrix tape. Columns of C3 are numbered to correspond with columns of B2 or, if B2 is omitted, with columns of A1.

Usage:

(1) Calling sequence is CALL CHAIN (13, 3).

(2) This subprogram uses both scratch tapes for intermediate data storage.

(3) The C3 matrix will be listed by column and will contain elements for every row of the A1 matrix.

(4) If an extra row is included in the B2 matrix, a comment is printed. This element is ignored, and calculations are continued.

(5) Maximum A1 matrix size is 6666 x 6666 with a 500 limit on the number of columns in the B matrix unless B is omitted. Size limits depend on the average bandwidth by N = 20000/(3+b) where A1 is the average number of elements in the band not including the diagonal. The maximum matrix size is attained, for a given problem, by numbering the grid-points so that the band is kept to a minimum and increased as slowly as possible. The program is intended to handle the solution of simultaneous equations for structures with stiffness matrices of order 2 x 2 to 500 x 500.

Coding Information:

(1) This link includes a driving program and special subroutine as follows:

| Label | Name | Purpose |
|---|---|---|
| M 0 | DRIVER | Main Program |
| M 1 | Subroutine CHOL | Directs equation solution |
| M 2 | Subroutine CODTAB | Forms table of matrix codes |
| M 3 | Subroutine CHOL 1 | Solve equations |
| M 4 | Subroutine BADIBM | Indicates machine fault |

The subroutine with calling arguments is called by:

CALL CHOL1 (A, N, M1)
A is the variable band array
N is the matrix order
M1 is the number of column vectors in B2

(2) The link requires additional SAMIS subroutines as follows:

Auxiliary Subroutines

| | |
|---|---|
| Z 7 | TAPES |
| Z 8 | COINS |

(3) The non-corrective exit is taken if A1 is not positive definite or no tape assignment is given for C3 as well as for format, shape, and listing errors.

(4) The link designation is (13, 0, 0, 1).

Operation Link: CODE, Matrix coder

Purpose: To put a precoded A1 matrix in coded format as the C3 matrix or rename a coded matrix as C3.

Restrictions:

(1) A1 and C3 must be assigned to tapes and, unless only one block long, on different tapes.

(2) All codes from the precoded matrix must fit in core.

(3) C3 will be in the same sort as A1.

Method: If already in coded format A1 matrix is read from tape, renamed, and written on the output matrix tape, one block at a time. If not, all codes are read into core. Elements are read into core one block at a time. Codes are generated for only the non-zero elements from the row and column designations. The C3 matrix is written on tape by block.

Usage: Calling sequence is CALL CHAIN (6, 4).

Coding Information:

(1) This link includes a driving program and a special subroutine as follows:

| Label | Name | Purpose |
|---|---|---|
| F 0 | DRIVER | Main Program |
| F 1 | Subroutine CODE | Directs calculations |

The calling sequence for CODE has no arguments.

(2) This link requires additional SAMIS subroutines as follows:

| Auxiliary Subroutines | | Other Subroutines | |
|---|---|---|---|
| Z 7 | TAPES | M 4 | BADIBM |
| Z 8 | COINS | Z 6 | TSNAM |

(3) It will be desirable to include the capability of in-core operation in this link to increase execution speed if extensive use is made of this link.

(4) The link designation is (6, 0, 0, 1).

Operation Link: COLS, Matrix Column Lister

Purpose: To resort coded A1 matrix, if necessary, so it is coded and listed by column as C3 and to column list a series of matrices.

Restrictions:

(1) The first assigned scratch tape cannot be used for A1 or C3.

(2) A1 and C3 may be assigned to core or tape. If A1 or C3 will not fit in core they must be assigned to a tape or tapes.

(3) For serial operation, the $C3_i$ are the column listed $A1_i$; $i = 1, N$.
The $A1_i$ and $C3_i$ must be sequential groups and $A1_i$ and $C3_i$ on different tapes.

Method: If the matrix is already column listed a comment is printed and the matrix written on tape if required. Otherwise, the row and column designations are interchanged in the codes and the matrix sorted. After sorting the codes are returned to their original status. In serial execution, tape and matrix numbers are increased by one, and the link re-executed.

Usage:

(1) The calling sequence is CALL CHAIN (3, 4).

(2) The E field should contain N, the number of times to repeat the instruction serially, in the form 100 N. If N = 0, it is assumed to be 1.

Coding Information:

(1) This link is contained within the SORT link. See the SORT writeup for additional coding information.

(2) The link designation is (3, 0, 4, 1).

Operation Link: DECO, Matrix Decoder

Purpose: To put the A1 matrix in precoded format as C3 or rename a precoded matrix.

Restrictions:

(1) A1 and C3 must be assigned to tapes and, unless only one block long, on different tapes.

(2) All codes of the precoded matrix must fit in core at once.

(3) C3 will be in the same listing as A1.

Method: If already precoded, the matrix is read from tape, renamed, and written on the output matrix tape. Alternately, the coded matrix, A1 is read from tape, a block at a time, a table of unique row and column codes is formed and written on the C3 matrix tape. The matrix is then reread and elements of the precoded matrix stored in blocks and written on tape. Zero elements are supplied as required. If an element is exactly 1.0E-30, it is replaced by zero. (This option is provided to permit reading a coded matrix with a null row or column and retaining the vector in the precoded format). In serial execution, the tape and matrix numbers are increased by one and the link re-executed.

Usage:

(1) Calling sequence is CALL CHAIN (7, 4).

(2) The E field should contain N, the number of times to repeat this instruction serially, in the form 100 N. If N = 0, it is assumed to be 1.

Coding Information:

(1) This link includes a driving program and a special subroutine as follows:

| Label | Name | Purpose |
|---|---|---|
| G 0 | DRIVER | Main Program |
| G 1 | Subroutine DECO | Directs Calculations |

The calling sequence for DECO is CALL DECO.

(2) This link requires additional SAMIS subroutines as follows:

| Auxiliary Subroutines | | Other Subroutines | |
|---|---|---|---|
| Z 7 | TAPES | M 4 | RADIBM |
| Z 8 | COINS | Z 6 | TSNAM |

(3) It will be desirable to include the capability of in-core operation in this subprogram if extensive use is made of this link.

(4) The link designation is (7, 0, 0, 1).

Operation Link: FLIP, Matrix Transposer

Purpose: To form C3, the transpose of the A1 matrix

Restrictions:

(1) A1 must be coded and row listed, column listed or unsorted. C3 will be coded and column listed, row listed, and unsorted, respectively.

(2) Blocks of A1 are treated one at a time.

Method: Row and column code designations are exchanged and listing indicator changed. Thus if A is originally row listed, $C = A^T$ is column listed. Modified blocks are then output as C3.

Usage: Calling sequence is CALL CHAIN (3, 4).

Coding Information:

(1) This link includes a driving program and special subroutines as follows:

| Label | Name | Purpose |
|---|---|---|
| E 0 | DRIVER | Main Program |
| E 1 | Subroutine FLIP | Directs Calculations |

The calli~ sequence for FLIP is CALL FLIP.

(2) This link requires additional SAMIS subroutines as follows:

| Auxiliary Subroutines | | Other Subroutines | |
|---|---|---|---|
| Z 7 | TAPES | H 2 | SWITCH |
| Z 8 | COINS | Z 6 | TSNAM |

(3) The link designation is (5, 0, 4, 1).

Operation Link: INKS, Matrix Printer and Puncher*

Purpose: To print out and optionally punch the A1, B2, and/or C3 matrices as required, in either coded or precoded format with appropriate heading data and to print series of matrices.

Restrictions:

(1) Matrices must be coded or precoded. They may be in any listing or sort. Precoded matrices must fit in core.

(2) All heading cards are printed before printing the first matrix. The first card is printed on each page for each matrix.

Method: All in-core matrices to be printed are printed first. Then all on-tape matrices required are printed. For precoded matrices the entire matrix is read into core a block at a time and then printed. Pre-coded matrices are partitioned vertically. Each printed page contains the matrix name, matrix heading, sequential page number, and appropriate row and column headings. If heading cards are introduced, the first card is used as the matrix heading. If no heading is provided, the matrix heading indicates the pseudo instruction number. The matrix end is indicated on the last page. Printing and punching of a sequence of matrices is also provided by recycling of the link. Matrices may also be punched.

Usage:

(1) Calling sequence is CALL INKS.

(2) The "Header Card" format is 12A6. Information in card columns 1 to 72 inclusive is printed.

(3) Let the E field input be defined as 100000L + 100N + M.
Then:
M is the number of header cards ($\leqslant$99)
N is the number of sequential matrices ($\leqslant$999)
L indicates the punching option. If L is = 0, the matrix only printed; if = 1, it is punched and printed.

Coding Information:

(1) This link includes a driving program and a special subroutine as follows:

* Note that punching may be inadmissible on some systems.

| Label | Name | Purpose |
|---|---|---|
| R 0 | DRIVER | Main Program |
| R 1 | Subroutine INKS | Direct printing |

The calling sequence for INKS is CALL INKS.

(2) This link requires additional SAMIS subroutines as follows:

| Auxiliary Subroutines | | Other Subroutines | |
|---|---|---|---|
| Z 7 | TAPES | M 4 | BADIBM |
| Z 8 | COINS | | |

(3) This link double spaces the matrix printout. Comment cards indicate the location of changes necessary to single space the matrix printout.

(4) Data in the DAT region, beginning at the bottom, is destroyed by a precoded matrix read from tape. The precoded matrix will occupy the first locations.

(5) If the ID of an on-tape matrix to be printed does not agree with the ID of the matrix on the pseudo instruction card, an error printout is made. If the precoded matrix will not fit in core an error comment is printed.

(6) The link designation is (18, 0, 7, 0).

Operation Link: ITER, Simultaneous Equation Solver by Accelerated Seidel Iteration.

Purpose: To form $C3 = A1^{-1} * B2$ where A1 is a square, positive-definite coded matrix, B2 a coded set of vectors, and C3 a coded matrix.

Restrictions:

(1) A1 must be square and have non-zero diagonal elements. The iteration will not converge if A1 is indefinite. A1 may be larger than core.

(2) A1 must be row listed and B2 column listed.

(3) C3 will be written in coded format, listed by column.

(4) A1, B2 and C3 must be assigned to different tapes. One column of C3 must fit in core. A1 and B2 may be smaller or larger than core.

(5) No two row numbers can be the same in absolute value. No more than 480 negative gridpoints can be used, with up to 10 components per point.

Method: Core storage is reassigned and tapes repositioned if necessary. As much of A1 as will fit is read into core, recoded, and diagonals checked to be non-zero. The rest of A1 if any, is similarly treated by blocks and put on the scratch tapes. The first block of B2 is read in. Rows corresponding to the first B2 block are relaxed and then additional B2 blocks are read and rows relaxed until the first column of the C3 matrix has been changed. When either the required accuracy or maximum number of iterations has been obtained, the columns are written in blocks on the output tape with column coding matching the B2 column designations. At the end of each iteration the number of significant figures of accuracy is indicated by the sense lights in binary.

A null vector is taken as the initial guess and standard Seidel iteration is used for the first relaxation cycle. Successive iterations are performed using an adjusted overrelaxation factor. For the first column of C3 this is defined as 1.40. When at least ten percent of the iterations have been performed, a new estimate for the overrelaxation factor is formed using Young's formula ("Iterative Methods for Solving Partial Difference Equations of the Elliptic Type", Trans. Amer. Math. Soc. Vol 76, p. 92). Successive cycles of iteration are performed using the estimate of the optimum relaxation factor. Options permit introducing the starting vector and relaxation factors and saving solution estimates every five minutes.

Usage:

(1) The calling sequence is CALL CHAIN (11, 4).

(2) Printout of iteration progress may be obtained by using a negative E field input in the pseudo instruction.

(3) If a number is included in the E field, its upper digit defines the number of cycles of iteration divided by 10. The last digit defines the iteration option. If the upper digit is zero, 3 significant figures are required. If the next 3 digits are zero, 9990 iterations are tolerated. The lower digit provides the following options:

a. Use null vector originally.
b. Read initial vector from C3 tape and first cycle and successive cycles relaxation factors (as a matrix) from input tape.
c. Save most recent iteration vector on C3 tape every five minutes.
d. Use previously generated data from assigned scratch tapes for matrix part that will not fit in core.

These options are selected on the basis of the value of the last digit in the E field as follows:

| Digit | a | b | c | d |
|---|---|---|---|---|
| 0 | Yes | No | No | No |
| 1 | Yes | No | Yes | No |
| 2 | No | Yes | No | Yes |
| 3 | No | Yes | Yes | Yes |
| 4 | Yes | No | No | Yes |
| 5 | Yes | No | Yes | Yes |

(4) Both scratch tapes are used by the program even though matrices may fit in core.

Coding Information:

(1) This link includes a driving program and special subroutines as follows:

| Label | Name | Purpose |
|---|---|---|
| K 0 | DRIVER | Main Program |
| K 1 | Subroutine ITER | Directs calculations |
| K 2 | Subroutine RECODE | Recodes the matrix and checks for null diagonals |

Calling sequences with arguments are:
CALL RECODE (L, JL, KA, JB)
L is the array name of the data to be coded.
JL is the last index of the array to be coded.
KA is the array of the unique codes
JB is the first index of code array to be used.

(2) This link requires additional SAMIS subroutines as follows:

| Auxiliary Subroutines | | Other Subroutines | |
|---|---|---|---|
| Z 7 | TAPES | M 2 | CODTAB |
| Z 8 | COINS | M 4 | BADIBM |

(3) Maximum matrix size is 20,000 x 20,000.

(4) Column codes of the C3 matrix are chosen to correspond with those of B2.

(5) If convergence is not obtained in the permitted number of cycles, the error exit is taken. This exit is also taken if a diagonal is zero or no tape assignment is given for output.

(6) Item (3)c, page 91, requires availability of a computer core storage clock to calculate iterative checkpoint times.

(7) The link designation is (11, 0, 0, 1).

Operation Link: MULT, Matrix Multiplier

Purpose: To form C3 = A1 * B2, and to multiply series of matrices.

Restrictions:

(1) A1 must be row listed and B2 column listed unless both are precoded. If precoded, each can have either listing.

(2) Either A1 or B2 must fit in core if coded. Both must fit in core if precoded.

(3) C3 must be written on tape, if coded. It may be on the same tape as A1 or B2. The first assigned scratch tape must not be used by A1, B2, or C3 since it may be required for intermediate data storage.

(4) The listing of C depends on whether or not A is in core during the multiplication, if coded. If precoded, C is row listed. If A1 does not fit in core B2 must be on a different tape, if taped.

(5) Serial multiplications include the following options:

$C3_i = A1 * B2_i$ for $i = 1, 2, \ldots N$ if A1 is indicated to be in core.

$C3_i = A1_i * B2$ for $i = 1, 2, \ldots N$ if A1 is on tape and B2 in core.

$C3_i = A1_i * B2_i$ for $i = 1, 2, \ldots N$ if both A1 and B2 are indicated to be on tape. In any event, A1 , B2 and C3 must be in sequential groups on each tape.

Method: If A1 and B2 fit in core and are coded, A1 is multiplied by B2 and the resulting matrix listed by column on the matrix output tape. If A1 is in core and B2 does not fit, A1 is multiplied by B2 and the partitioned results written on scratch tape. SORTS is then used to sort the complete C3 matrix by column and write the result on the matrix output tape. If A1 does not fit in core, the partitioned C3 matrix is written on the scratch tape and SORTS used to sort C3 and list it by row. If both matrices are precoded, both are read into core and the precoded matrix C3 written on tape directly. Serial operation involves recycling link instructions after incrementing tape group and matrix numbers.

Usage:

(1) Calling sequence is CALL CHAIN (9, 4).

(2) This subprogram may use the first assigned scratch tape for intermediate storage.

(3) Information in the E field of the pseudo instruction defines which of the A1 or B2 matrices is larger than core (saving machine time) or the number of serial multiplications as follows:

Let the E field be defined as 100 N + L. Then N is the number of serial operations and L is interpreted as follows:

0: Undefined - either matrix may exceed size.
1: The A matrix is larger than core.
2: The B matrix is larger than core.
>2: Undefined

Coding Information:

(1) This link includes a driving program and a special subroutine as follows:

| Label | Name | Purpose |
|---|---|---|
| I 0 | DRIVER | Main Program |
| I 2 | MULTPY | Multiplies precoded matrices |
| I 3 | Subroutine MULT | Directs matrix multiplication |

The calling sequence for MULT is CALL MULT.

(2) This link requires additional SAMIS subroutines as follows:

| Auxiliary Subroutines | | Other Subroutines | |
|---|---|---|---|
| Z 7 | TAPES | N 1 | SORTS |
| Z 8 | COINS | N 2 | MIXES |
| | | N 3 | ORDER |
| | | M 4 | BADIBM |
| | | H 2 | SWITCH |
| | | Z 6 | TSNAM |

(3) The link designation is (9, 0, 6, 1).

Operation Link: READ, Matrix Card Reader

Purpose: To read matrices from the input tape and store them in core or on tape and to read series of matrices.

Restrictions:

(1) The identification card must precede the matrix and matrix data must correspond with identification information.

(2) The matrices must be in the sequence in the card deck in which they are called by the pseudo instruction (A1,B2,C3,F1 . . .).

(3) Matrix data may be in coded or precoded format. If coded, a matrix may be in row, column, unsorted listing; if precoded it may be in row or column listing.

(4) In serial operation, the $A1_i$, $B2_i$ and $C3_i$ matrices are read where $i = 1, 2, 3 \ldots N$. Each series (e.g., $A1_i$) must be assigned to tape with sequential group numbers. $A1_i$, $B2_i$, and $C3_i$ must be assigned to different tapes.

Method: The identification card is read. The matrix number is checked with that required and the matrix data read using the identification information. The matrix is read and stored in core or on tape one block at a time. In serial execution, the tape and matrix numbers are augmented by one and the link re-executed.

Usage:

(1) Calling sequence is CALL CHAIN (1, 4).

(2) The identification card must be in format (A3, I3), 5I6. It contains the following data:

(A3, I3): The alphanumeric identification of the matrix. The numeric designation must be greater than zero.

(I6): The number of input cards in the matrix including codes.

(I6): The number of rows in the matrix. } may be zero for coded matrices

(I6): The number of columns in the matrix. }

(I6): Listing: -1, listed by row; 0, unsorted; +1, listed by column.

(I6): Format: 0, coded; 1, precoded.

(3) Coded matrices must be in format (3(2(I5, I1),E12.0)). Data on each card is interpreted as follows:

(I5,I1): The row identification number

(I5,I1): The column identification number

(E12.0): The matrix element

(4) Precoded matrices must have row and column codes in format (12(I5, I1)) and elements in format (6E12.0). Codes must precede matrix elements. The first matrix element must start a new card. If row listed, column codes precede row codes and vice versa. In any event, the second set of codes must begin right after the first set is completed. Similarly, the matrix elements must be in sequence with no omissions.

(5) If the E field input is negative, card data will be printed as it is read.

(6) E field input is | 100 N| where N is the number of serial operations. If N = 0 it is assumed to be 1.

Coding Information:

(1) This link requires a driving program and two special subroutines:

| Label | Name | Purpose |
|---|---|---|
| A 0 | DRIVER | Main Program |
| A 1 | Subroutine READ | Directs data reading |
| A 2 | Subroutine CR | Stores matrix in core |

The calling sequence with arguments is:

CALL CR (LAST, LA, NDIM, ELE, INE)
LAST indicates block number being moved.
LA is the index of the next location to which data is to be moved.
NDIM is the number of core locations remaining for storing matrix data.
ELE is the array to be moved into core.
INE locates the identification block in TRA for the matrix being stored.

(2) This link requires additional SAMIS subroutines as follows:

| Auxiliary Subroutines | | Other Subroutines | |
|---|---|---|---|
| Z 7 | TAPES | M 4 | BADIBM |
| Z 8 | COINS | | |

(3) The non-corrective error exit is taken if identification does not match that of the pseudo card or if a zero code is used in a precoded matrix. Format errors will also occur if cards are improperly punched.

(4) The link designation is (1, 0, 0, 3).

Operation Link: ROOT, Symmetric Matrix Root Extracter

Purpose: To obtain the latent roots C3 and vectors B2 of A1, where A1 is a symmetric precoded matrix.

Restrictions:

(1) The A1 matrix must be in precoded format, and prestored in the first part of the data region (by MINTS). A1 may be row or column listed.

(2) Matrix order must be greater than 0 and less than 131.

(3) The eigenvector matrix, B2 will be precoded and column listed and vectors are columns of the matrix.

(4) The eigenvalue matrix, C3 will be column listed and coded.

Method: Jacobi's method (see Modern Computing Methods, Philosophical Library, 1961, p. 30, 31) is used. Matrix shape, size, format, and symmetry are checked. Symmetry is required to four significant figures. Up to 40 nonsymmetric elements will be indicated but calculations continue. If the matrix is too large, the recoverable error exit is taken. If the roots can be obtained, the matrix is expanded into a packed upper triangular array and a JPL FORTRAN subroutine is used to get both the roots and vectors. Vectors are given the B2 matrix designation; and roots, the C3 designation. Vectors are of unit length. Row codes of the B2 matrix are taken to be those of the A1 matrix. Column codes of the B2 and C3 matrices are numbered in sequence. B2 is written in precoded format on tape, if required. B2 is column listed, each column containing a modal vector.

Usage:

(1) Calling sequence is CALL CHAIN (15, 4).

(2) This subprogram uses JACOBI (JPL FORTRAN subroutine) as a subroutine.

(3) The E field information, N, defines the number of roots to be obtained and directs printout. If $N = 0$ or $|N| \geq |M|$, the matrix order, all roots and vectors appear in C3 and B2. Otherwise $|N|$ roots and vectors appear in C3 and B2. If $N < 0$, the eigenvalues in C3 are printed out by ROOT.

Coding Information:

(1) This link includes a driving program and a special subroutine:

| Label | Name | Purpose |
|---|---|---|
| 0 | DRIVER | Main Program |
| 1 | Subroutine ROOT | Directs Eigenvalue Calculations |
| 3 | Subroutine JACOBI | Eigenvalue Program |

Calling sequences have no arguments.

(2) This link requires additional SAMIS subroutines as follows:

| Auxiliary Subroutines | | Other Subroutines | |
|---|---|---|---|
| Z 7 | TAPES | M 4 | BADIBM |
| Z 8 | COINS | | |

(3) To maximize the size of the matrix that can be handled, mapping of storage in this link is not restricted by the SAMIS DAT array size (20,000 locations).

(4) The link designation is (15, 4, 4, 2).

Operation Link: ROWS, Matrix Row Lister

Purpose: To resort a coded A1 matrix, if necessary, so it is coded and listed by row as C3 and to row list a series of matrices.

Restrictions:

(1) The first assigned scratch tape cannot be used for A1 or C3.

(2) A1 and C3 may be assigned to core or tape. If A1 and C3 will not fit in core, they must be assigned to a tape or tapes.

(3) For serial operation, C3 are the row listed $A1_i$; i = 1, 2, 3 . . . N. The $A1_i$ and $C3_i$ must be in sequential groups and $A1_i$ and $C3_i$ on different tapes.

Method: If the matrix is already row listed a comment is printed and the matrix written on tape if required. Otherwise, the row and column designations are interchanged in the codes and the matrix sorted. After sorting the codes are returned to their original status. In serial execution, tape and matrix numbers are increased by one, and the link re-executed.

Usage:

(1) The calling sequence is CALL ROW (4, 4).

(2) The E field should contain the number, N, of times to repeat this instruction serially in the form 100 N. If N = 0, it is assumed to be 1.

Coding Information:

(1) This link is contained within the SORT link. See the SORT writeup for additional coding information.

(2) The link designation is (3, 0, 4, 0).

**Operation Link:** SORT, Matrix Sorter

**Purpose:** To sort a coded matrix A1, eliminating null elements and elements with the same code and producing a row or column listed coded C3 matrix or to sort a series of matrices.

**Restrictions:**

(1) Neither the A1 nor the C3 can be assigned to the first assigned scratch tape.

(2) A1 and C3 may be assigned to core or tape. If A1 and C3 will not fit in core they must be assigned to a tape or tapes.

(3) For serial operation, C3 are the sorted $A1_i$, i = 1, 2, 3, . . . N. The $A1_i$ and $C3_i$ must be in sequential groups and $A1_i$ and $C3_i$ on different tapes.

**Method** If the matrix is in core or fits in core, it is sorted and written on tape, if required. If the matrix does not fit into core, core is filled and data sorted and the resulting partition written on the scratch tape by block. The remaining blocks are sorted and mixed so that after the entire matrix is passed over, the lowest element numbers remain in core. Each sorted block is also written on the scratch tape. The lowest "slab" in core is written from core onto the output matrix tape. The scratch tape is then read a multiple number of times, the slab of the next lowest codes and corresponding elements being extracted in each pass and written on the output tape. If serial execution is required, tape and matrix names are increased by one and the link recycled. If no tape assignment is given for A1, serial operation is omitted.

**Usage:**

(1) Calling sequence is CALL CHAIN (14, 4).

(2) This link may use the first assigned scratch tape for intermediate storage.

(3) Output listing is by row or column depending on the E field input. If $E < 0$, the result is row listed. If $E > 0$, the output is column listed.

(4) The E field contains the number, N, of times to repeat the instruction serially, in the form $\pm 100$ N. If N = 0, it is assumed to be 1.

Coding Information:

(1) This link includes special subroutines as given below. The driver program is combined with the COLS and ROWS driver programs.

| Label | Name | Purpose |
|---|---|---|
| N 1 | Subroutine SORTS | Directs calculations |
| N 2 | Subroutine MIXES | Augments a sorted set |
| N 3 | Subroutine ORDER | Orders a set of data by increasing code magnitude |

Calling sequences for subroutines are as follows:

CALL SORTS (M, N, NI)
M is the sorting indicator
-1, order codes (with elements)
0, interchange row and column numbers in codes before ordering
1, interchange row and column numbers in codes before and after ordering
N is the number of matrices involved (one or two).
If $N < 0$, -N is the number of blocks on the scratch tape and MULT is the operation.
NI is the union indicator
1, add matrices A and B
-1, subtract matrices A and B

CALL MIXES (KUT, KUP)
KUT is the highest code for which sorting is complete,
KUP is the index of the last code in the DAT region.
New codes may be stored two locations beyond this location upon entry to MIXES.

CALL ORDER (A, LA)
A is the array to be ordered
LA is the array index of the last code in the array

(2) This link requires the auxiliary subroutines Z7 TAPES and Z8 COINS.

(3) The link designation is (14, 0, 6, 1).

Operation Link: SUBS, Matrix Subtracter

Purpose: To form C3 = A1 - B2 where A1, B2 and C3 are coded matrices and to subtract series of matrices.

Restrictions:

(1) A1 and B2 must be in the same listing; either row or column. The C3 matrix will have the same listing as A1 and B2.

(2) A1 or B2 may be in core or both on tape. If C3 does not fit in core, a tape assignment must be specified for it.

(3) For serial operation ($C3_i = A1_i - B2_i$) i - 1, 2, 3, ... N where $A1_i$, $B2_i$, and $C3_i$ are assigned to tapes. Each series (e.g., $A1_i$) must be in sequential groups. $A1_i$, $B2_i$ and $C3_i$ must be on different tapes.

Method: The procedure is the same for matrix addition (ADDS). On the first pass through the B2 matrix however, signs of the elements are changed.

Usage:

(1) Calling sequence is CALL CHAIN (8, 4).

(2) The information in the E field of the pseudo instruction defines 100 N, for serial subtraction. If N = 0, it is assumed to be 1.

(3) When C3 is formed of more than one partition, the recoverable error exit (NCE = -1) is taken.

Coding Information:

(1) This link is contained in the ADDS link. For further coding information, see the ADDS writeups.

Operation Link: WASH, Special Matrix Multiplication

Purpose: To operate with a coded A1 matrix on a coded B2 matrix to delete or extract row, column, row and column, partitions or elements or scale elements of these partitions and output the result as a coded C3 array.

Restrictions:

(1) The A1 matrix must fit in core, be coded and row listed. B2 need not be sorted nor fit in core.

(2) C3 will be coded, in the same sort as B2 and may be left in core or put on tape. If C3 will not fit in core, it must be given a tape assignment.

Method: The A1 matrix is checked and if it is not diagonal, this is indicated. Blocks of B2 are treated, one at a time. The code of each element of B2 is decomposed into a row and column code. For each element of B2, A1 is searched to locate elements with corresponding codes. In accordance with the option selected, the B2 element is destroyed, retained, or scaled and written on tape or left in core as required.

Usage:

(1) Calling sequence is CALL CHAIN (10, 4).

(2) Operation is controlled by E field input. The zero option is as follows:

E field = 0: Performs the multiplication $A1 * B2 * A1^T$ where A1 is a diagonal matrix. Uses of the option are:

a. To delete particular rows and columns of B2 from the matrix. Elements of A1 are entered as zeros and row and column codes are chosen to correspond with B2 rows and columns to be deleted.

b. To scale elements in diagonal partitions of the B2 matrix.

c. To do a combination of a and b.

Options 1, 2, 3, and 4 modify each element of B2 depending on the first element of A1 (proceding through each row in sequence) with a matching row or column code.

E field = 1: If either row or column code match, the B2 element is multiplied by the corresponding A1 element. If both row and column code match, the B2 element is multiplied by the square of the A1 element. If no match of a B2 element is found in A1, the element is preserved unchanged. The uses of the option are:

a. To delete particular rows and columns of B2 from the matrix. Elements of A1 are entered as zeros and row and column codes are chosen to correspond with B2 rows and columns to be deleted.

b. To scale elements in diagonal partitions of the B2 matrix.

c. To scale elements in any particular partition of the B2 matrix.

E field = 2: Same as 1 but deletes all B2 elements for which no match is found in A1. Uses of this option are:

a. To extract particular rows and columns from B2. Elements of A1 are entered as ones and row and column codes chosen to correspond with B2 rows and columns to be extracted.

b. To extract particular rows or columns from B2. Elements of A1 are entered as one and row or column codes chosen which do not match any rows or columns of B2 to extract columns or rows respectively.

c. To do a combination of a and b.

E field = 3: If, for an element of B2 there is an element of A1 whose row and column code correspond, the B2 element is multiplied by the square of the A1 element. All other elements of B2 are preserved unchanged. Uses of this option are:

a. To delete particular elements from B2. Elements of A1 are entered as zeros and codes chosen to match those of the elements to be deleted.

b. To scale particular elements of B2.

c. To do a combination of a and b.

d. To square elements of a matrix. The A1 and B2 matrices are then the same.

E field = 4: Same option as when E field = 3 except non-matching elements are deleted. Uses of this option are similar to those of option 3.

Coding Information:

(1) This link includes a driver program and special subroutines as follows:

| Label | Name | Purpose |
|---|---|---|
| J 0 | DRIVER | Main Program |
| J 1 | Subroutine WASH | Directs Calculations |

The calling sequence for WASH is CALL WASH.

(2) This link requires additional SAMIS subroutines as follows:

| Auxiliary Subroutines | | Other Subroutines | |
|---|---|---|---|
| Z 7 | TAPES | M 4 | BADIBM |
| Z 8 | COLNS | Z 6 | TSNAM |

(3) If the B2 matrix will fit in core, the C3 matrix will always fit in core.

(4) In addition to format errors, it will be noted when A1 is not a diagonal matrix or if a zero code is introduced before the last element.

(5) A recoverable error is taken when A1 will not fit in core.

(6) The link designation is (10, 4, 6, 1).

# SECTION 7

# PARAMETRIC INPUT DATA

This section describes three principal types of input data: material tables, element data, and matrix data. Material tables define the mechanical properties of the structural materials. Element data define the local geometry (member thickness, cross-sectional areas, moments of inertia), gridpoints (numbers and locations), coordinate systems, temperatures, weight, and pressure on each structural element. Matrix data can be used to define input data to be used in matrix operations for such things as imposing holonomic boundary conditions, making equilibrium checks and introducing concentrated loadings. In addition, of course, matrix input can be used in matrix operations not associated with structural problems.

In addition to these types of input, there are three other types: link control data, pseudo instructions, and heading cards. Link control data tables are standardized. They are the first data in the deck and follow the "DATA" IBSYS control card. They are defined in Table 4-6. Pseudo instructions appear second in the data deck. Their formulation and form are described in Section 4. Heading cards are read by the INKS link. They are in alphanumeric format and are described in the INKS writeup.

For completeness, a brief description of FORTRAN formats is described in the next paragraphs. Description of the material and element data input is followed by a description of input data deck arrangement.

## 7.1 EXPLANATION OF FORTRAN FORMATS

I (Integer) Format: The form Iw specifies an integer field w columns wide, where "w" includes the digits, any blanks, and a sign. The integer value itself must be less than 32768. The field is always right justified, e.g., 4987 written in format I6 would appear as bb 4987 where b indicates a blank.

F (External Fixed Point) Format: The form Fw.d specifies a decimal field w columns wide, where "w" includes a sign, the digits, a decimal point, and any blank and "d" specifies the number of digits after the assumed decimal point. When F format is used for input, the decimal point is optional. However, if it is inserted, it overrides the "d" specification. Example: If the input is F8.4, the number -2.376 could be punched as -2.376. In this case, the three place decimal overrides the d = 4 specifications of the F field. 382 could be written as b 3820000 for format F8.4 with the decimal omitted.

In using an F format the size of the output numbers must be known so that the "w" specification will encompass at least the integer portion and sign of the number. Otherwise leading digits will be dropped.

E (Floating Point) Format: The form Ew.d specifies a floating point field w columns wide, where "w" includes a sign, the digits, a decimal point, the exponent sign, the exponent, and any blanks. The "d" specifies the number of digits after the decimal point, not counting the exponent and its sign. For input a decimal point is optional in the mantissa. If it is used, it overrides the "d" specification.

Example: E12.4, 498.72 could be punched on the card as 498.72E-02 or as 498.72E+00. The exponent cannot exceed ± 38 in magnitude. There are several ways to write E formats. A FORTRAN manual* should be consulted to take advantage of several shortcuts available.

A (Alphanumeric) Format: The form Aw specifies an alphanumeric field w column wide. The characters can be any of the characters acceptable to the computer including blanks. This is the format used to read in header cards, matrix identifications and instruction names.

X (Space) Format: The form wX specifies a field of w spaces (blanks) wide.

## 7.2 MATERIAL TABLES

Material tables define the mechanical properties of all materials of interest. When new materials are to be considered they can be added to the table. The only limitation on the number of materials that may be included is that the table can only involve a total of 398 cards.

Most material table data is entered as floating decimal input in format E8.4. Two cards must be entered for each material. Data can be entered in the table for as many temperatures as desired. For a given material, the cards must be sorted so that temperatures increase from

---

*Daniel D. McCracken, "A Guide to FORTRAN Programming," John Wiley & Sons, Inc., New York.
"Reference Manual 709/7090 FORTRAN Programming System," C28-6054-2 IBM Corporation

one pair to the next. If data is required at a temperature not given in the tables, then BILD will extrapolate or interpolate the material properties using linear interpolation and data for the two closest temperatures. If material properties are given for only one temperature and data is wanted at another temperature, BILD will use the material properties for the temperature appearing in the table. A single zeros card is used to signify the end of the table. This blank card is not counted in the above limitation of the 398 cards in the material table.

Table 7-1 defines the format, interpretation, and an example of the two cards required for each material entry. The material stiffness coefficients indicated enter into the stress-strain equations in accordance with the equations.

$$\begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \\ \sigma_{zz} \\ \sigma_{xz} \\ \sigma_{yz} \end{Bmatrix} = \begin{bmatrix} D_{11} & & & & & \\ D_{21} & D_{22} & & & & \\ D_{31} & D_{32} & D_{33} & & & \\ D_{41} & D_{42} & D_{43} & D_{44} & & \\ 0 & 0 & 0 & 0 & D_{55} & \\ 0 & 0 & 0 & 0 & D_{65} & D_{66} \end{bmatrix} \begin{Bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{xz} \\ \epsilon_{zz} \\ \epsilon_{xz} \\ \epsilon_{yz} \end{Bmatrix}$$

These define a "monotropic" material with at least one plane of elastic symmetry (the x-y plane). If values of $D_{31}$, $D_{32}$, $D_{43}$, and $D_{65}$ are assigned zero, an orthotropic material (three mutually orthogonal planes of symmetry) can be defined. If, in addition $D_{33} = D_{55} = D_{66}$, $D_{11} = D_{22} = D_{44}$, $D_{21} = D_{41} = D_{42}$ and $D_{21} = \frac{V}{1-V} D_{11}$ an isotropic material is described. Then, $D_{55} = \frac{E}{2(1+V)}$, $D_{11} = \frac{E(1-V)}{(1+V)(1-2V)}$, where E is Young's Modulus and V, Poisson's ratio. The elastic constants are referenced to the local coordinate axes of each element. Therefore, whenever a local

TABLE 7-1

MATERIAL TABLES INPUT DATA

| Card | Field | Format | Information |
|---|---|---|---|
| 1 | 1 | A2,A4) | The material identification: Each material must have a unique identification number. It is recommended that standard SAE and Aluminum Association numbers be used insofar as possible. Only the leading two characters of the six-character identification number are significant. |
| 1 | 2 | (E8.0) | Rankin temperature for the given material properties. |
| 1 | 3 | (E8.0) | Coefficient of thermal expansion: inches/inch-degrees Rankin. |
| 1 | 4-9 | (6E8.0) | Material stiffness coefficients; $D_{11}$, $D_{21}$, $D_{22}$, $D_{31}$, $D_{32}$, $D_{33}$, lbs/sq.in. |
| 2 | 2-8 | (7E8.0) | Material stiffness coefficients; $D_{41}$, $D_{42}$, $D_{43}$, $D_{44}$, $D_{55}$, $D_{65}$, $D_{66}$, lbs/sq.in. |

SAMPLE MATERIAL TABLE INPUT

(Isotropic Aluminum)

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| FORMAT | (2X,A2,A4) | (E8.0) | (E8.0) | (E8.0) | (E8.0) | (E8.0) | (E8.0) | (E8.0) | (E8.0) |
| CARD 1 | 2014T6 | .53E+3 | .125E-4 | 16.E+6 | 8.E+6 | 16.E+6 | 0 | 0 | 4.E+6 |
| CARD 2 | | 8.E+6 | 8.E+6 | 0 | 16.E+6 | 4.E+6 | 0 | 4.E+6 | 0 |
| CARD 3* | 00000000 | 00000000 | 00000000 | 00000000 | 00000000 | 00000000 | 00000000 | 00000000 | 00000000 |

* Used to indicate that all pairs of material table cards, for all materials, have been read.

coordinate system is selected for an element, the elastic constants must be input for that set.

For the Facet element, the x-y plane must be made to coincide with the midplane of the surface. For the line element, the x axis coincides with the line. The y and z axes locations are defined by the element data.

## 7.3 ELEMENT DATA

The purpose of the element data is to identify the element and define its state and geometry. Element data consists of at least two cards and at most nine cards of input per element. The two required cards are the first and ninth. The first card contains gridpoint numbers and local geometry data. The ninth card defines the coordinates of the first, second, and third gridpoints of the element. The other cards define the continuity conditions, weight, pressure and temperatures, local coordinate system description, and substitute and other gridpoint data.

No machine check is made between blocks of element data to insure data consistency. For an element, however, data is checked to make certain that a practical structure is being defined. This checking consists of insuring that quantities such as pre-integrated geometry (e.g., thickness, moment of inertia, and cross-sectional area), are positive or zero entities, and sufficient non-zero data is included to define a structure. If zero member properties are used, a comment is printed to alert the analyst to degenerate element stiffnesses.

In analyzing a structure, the analyst is concerned with a coordinate system for four purposes: 1) He chooses to describe the geometry of the element by choosing coordinates in some coordinate system. 2) At each gridpoint, he chooses a coordinate system to which his displacements are referred. 3) He chooses a coordinate system to which his elastic constants are related. 4) He chooses coordinates to which stress resultants are referred.

In SAMIS, all these coordinate systems must be right-handed, rectangular cartesian systems. Moreover, to reduce the number of coordinate systems, the analyst is limited to selection of at most two coordinate systems for each structural element.

For each element, the analyst can use the overall coordinate system and a local system. The overall system is selected by the user. The local system is defined by the user or, if not defined by the user is defined by the program. The user defines the local system by locating its axes with respect to the overall coordinate system. This data is contained on card 5 of the element data (see Tables 7-2 and 7-3). It includes the location of the origin of the local coordinate system, location of a point on the local x axis, and location of a point in the local x-y plane. These points are all defined by their coordinates in the overall system. The program defines the local z axis by taking the cross-product of the vector along the local x axis and a vector running from the origin of the local axes to the given point in the local x-y plane. The y axis is defined such that the x, z and y axes define a right-handed coordinate system.

If the local system is not defined by the user and is required, it is generated by the program. This nominal local coordinate system is defined for each element type. Thus, the line element is chosen to have its local

TABLE 7-2

ELEMENT DATA FORMAT FOR FACET

| CARD NO. | ELEMENT NUMBER | ELEMENT IDENTITY | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | | Elastic Node No.1 | Elastic Node No.2 | Elastic Node No.3 | | Substitute Node No.1 | Substitute Node No.2 | Substitute Node No.3 | Weight | Thickness | Material Identity |
| 2 | | | Continuity Node No.1 | Continuity Node No.2 | Continuity Node No.3 | | Pressure | Material Temperature | Temperature Change Upper Surface | Temp. Change Lower Surface | | |
| 3 | | | NOT | USED | WITH | FACET | | | | | | |
| 4 | | | NOT | USED | WITH | FACET | | | | | | |
| 5 | | | Coord. $X_1$ | First Local $Y_1$ | Gridpoint $Z_1$ | Coord. $X_2$ | Second Local $Y_2$ | Gridpoint $Z_2$ | Coord. $X_3$ | Third Local $Y_3$ | Gridpoint $Z_3$ | |
| 6 | | | NOT | USED | WITH | FACET | | | | | | |
| 7 | | | Coord. $X_1$ | First Subs. $Y_1$ | Gridpoint $Z_1$ | Coord. $X_2$ | Second Subs. $Y_2$ | Gridpoint $Z_2$ | Coord. $X_3$ | Third Subs. $Y_3$ | Gridpoint $Z_3$ | Coord. Identity |
| 8 | | | NOT | USED | WITH | FACET | | | | | | |
| 9 | | | Coord. $X_1$ | First Elastic $Y_1$ | Gridpoint $Z_1$ | Coord. $X_2$ | Second Elastic $Y_2$ | Gridpoint $Z_2$ | Coord. $X_3$ | Third Elastic $Y_3$ | Gridpoint $Z_3$ | Coord. Identity |

TABLE 7-3

ELEMENT DATA FORMAT FOR BEAM

| CARD NO. | ELEMENT NUMBER | ELEMENT IDENTITY | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | | Elastic Node No.1 | Elastic Node No.2 | Elastic Node No.3 | Cross-Sectional Area | Polar Moment of I | Shear Area Y Force | Moment of I About Z Axis | Shear Area Z Force | Moment of I About Y Axis | Material Identity |
| 2 | | | Continuity Node No.1 | Continuity Node No.2 | | Weight | Pressure | Material Temperature | Temperature Change From Zero Str. | Temperature Gradient X-Y Direction | Temperature Gradient X-Z Direction | |
| 3 | | | NOT | USED | WITH | BEAM | | | | | | |
| 4 | | | NOT | USED | WITH | BEAM | | | | | | |
| 5 | | | Coord. First Local Gridpoint $X_1$ | $Y_1$ | $Z_1$ | Coord. Second Local Gridpoint $X_2$ | $Y_2$ | $Z_2$ | Coord. Third Local Gridpoint $X_3$ | $Y_3$ | $Z_3$ | |
| 6 | | | NOT | USED | WITH | BEAM | | | | | | |
| 7 | | | Substitute Node No.1 | Substitute Node No.2 | | Coord. First Substitute Gridpoint $X_1$ | $Y_1$ | $Z_1$ | Coord. Second Substitute Gridpoint $X_2$ | $Y_2$ | $Z_2$ | Coord. Identity |
| 8 | | | NOT | USED | WITH | BEAM | | | | | | |
| 9 | | | Coord. First Elastic Gridpoint $X_1$ | $Y_1$ | $Z_1$ | Coord. Second Elastic Gridpoint $X_2$ | $Y_2$ | $Z_2$ | Coord. Third Elastic Gridpoint $X_3$ | $Y_3$ | $Z_3$ | Coord. Identity |

x axis along the line and its x-y plane defined by this axis and the third gridpoint used for the beam to define the principal plane. The local axis for the Facet is chosen to have its origin at the first gridpoint input for the Facet; its x axis being along the line joining the first two gridpoints input, and its x-y plane coincident with the plane of the Facet. The direction of the local z axis is defined by the order in which gridpoints are listed using the right-hand rule.

Table 7-4 summarizes the choices the user has for coordinate systems. He may define the coordinates of the gridpoints of the element in either the overall or a local coordinate system. If he chooses the local system, however, he must provide card 5 of the element data with the element involved. In giving the coordinates of the gridpoints, the analyst defines whether all the gridpoints and coordinates of an element are in the local or the overall system. This flag immediately follows the coordinates on cards 7 and 9 (see Tables 7-2 and 7-3).

Gridpoint displacements may be obtained in either the overall or local system. If Facets are to be used and the displacements are to be obtained in the local system, card 5 must be introduced to insure that joints common to two or more Facets are referenced to the same coordinate system. If card 5 is used with the set, then this input overrides the internally generated local system. A comment is printed if the z axis is not normal to the Facet plane or line element axis not coincident with one of the local coordinate axes. Displacements and gridpoint forces are related to the local coordinate system if the gridpoint number is negative. This device permits the user to choose to use different coordinate systems at gridpoints of an element. For example, displacements may be obtained in a local system at the first gridpoint while the overall basis is used for the second.

TABLE 7-4

COORDINATE SYSTEM CHOICE

| Purpose | Overall | Local | Card 5 Req'd if Local? |
|---|---|---|---|
| Define Geometry | Yes | Yes | Yes |
| Gridpoint Displacements | Yes | Yes | No* |
| Stress Reference | No | Yes | No |
| Monotropy Reference | No | Yes | No |

*If Facets are to be used and the displacements are to be obtained in the local system, card 5 must be introduced to insure that joints common to two or more Facets are referenced to the same coordinate system.

The user can select coordinate systems defining geometry indiscriminately, however, he cannot select the displacement coordinate system at each gridpoint indiscriminately. To combine equations for two elements meeting at a gridpoint, the equations must be in a common coordinate system for that gridpoint. If the overall system is used at all gridpoints, the analyst cannot make an error. If the local system is used, all elements coming into a common gridpoint must have their displacements in parallel local systems. Local x, y, and z axes must be positive in the same directions. This is guaranteed by providing card 5 for each element involving the common gridpoint and using the same system for each element. If the internally generated local system is used, however, the user must take care to insure that all elements coming into a gridpoint have a common x-y plane. Since the z axis is defined from the cross-product, common positive direction for the z axis depends on the location of the three points defining the x-y plane. For the Facet, listing element gridpoints consistently in counter-clockwise order when viewing the plane always from the same side insures a common z axis direction.

The coordinate system to which the stress is related must be a local system. This will be the system defined by the data on card 5. If this data is omitted, stresses are referenced to the nominal system generated by the program.

Similarly, the axis to which the material elastic coefficients are related must be a local system, which may either be input or program generated. However, a limited transformation is applied to the elastic coefficients so that further restrictions are imposed on the choice of local coordinates if the material is not isotropic. For the Facet, the plane of symmetry to which the material is related is defined to be coincident with the Facet midplane. For the line element, the material x axis coincides with the line and the y and z axes are taken to be the internally generated local axes. These limitations are used because of the infrequency with which material monotropy is used. Note that the limitations do not restrict the capability to treat monotropic materials since local axes can always be chosen for material representation only or material coefficients introduced with respect to any axis.

In reviewing the choice of coordinates available to the analyst, it can be seen that complete freedom of choice is not available. The coordinate systems introduced are limited, but a given coordinate system can be used for several purposes. The overall system may be used to define geometry and/or gridpoint displacement coordinates. A single local system may be the basis for geometry, displacements, stresses, and monotropy coordinates.

Despite the extra effort involved in using local coordinates, it is often desirable. For the Facet, for example, no degree of freedom corresponding to rotation about the local z axis due to a couple exists if the displacements are retained in the program generated local system, (or any system with an axis normal to the Facet plane). This reduces the number of equations generated.

If desired, the user can include substitute gridpoints in his analysis. Whereas elastic gridpoints are used to define the load-deflection relations for an element, the substitute gridpoints provide for transmitting load to an element at other than the elastic gridpoints. One substitute gridpoint can be chosen corresponding to each elastic gridpoint. Substitute points may be chosen anywhere and be defined in either the overall or a local coordinate system. Load transfer occurs at the substitute gridpoint. Physically, load can be visualized as being carried from the substitute gridpoint to the elastic gridpoint through a rigid link.

The analyst can introduce gridpoint discontinuity numbers to indicate special characteristics of the connection of each structural element gridpoint to its joint of the same number. The numbers define discontinuity between the gridpoints, and joints used in the analysis. Thus, if there are no substitute gridpoints, the discontinuity numbers pertain to the elastic gridpoints. If there are substitute gridpoints, the discontinuity numbers pertain to the substitute gridpoints. Discontinuity numbers are given in the same order as the analysis gridpoint numbers. Thus, the first seven-column field of discontinuity numbers relates to the first analysis gridpoint given, the second field to the second and so forth.

The discontinuity number is a right adjusted six digit number in a seven column field. Each digit corresponds to one component of displacement. The first digit corresponds to the deflection in the x direction; the second, the deflection in y; the third, deflection in the z direction. The fourth, fifth, and sixth correspond to rotation about the x, y, and z axes respectively. The coordinate axes referred to are either the local or overall axes depending on gridpoint number. If the number is positive, overall displacement coordinate axes are referenced. If the gridpoint number is negative, local gridpoint coordinates are referenced.

If the deformation component of the gridpoint is to be made compatible with that of the joint, the pertinent component digit may be left blank or punched as zero. Thus, in the usual case where there is no discontinuity for any deformation component at a joint, the discontinuity number would be b000000, or left blank (where b indicates an omitted punch (blank)).

If the deformation component of the gridpoint is to be distinct from that at the joint, the pertinent component digit may be assigned a number from one to eight. The effect of this input is to define an independent degree of freedom at the joint for the indicated displacement component. Therefore, the discontinuity number b000100 indicates that the gridpoint has a hinge, between the element gridpoint and the joint, which permits rotation about the x axis. The discontinuity number b000111 would model a ball bearing joint.

The digits one through eight are added to the gridpoint number to define a new joint number. The analyst must select the digit so that he does not direct the computer to develop a joint number which corresponds with one not intended to be connected with the discontinuity degree of freedom. Since joints numbers need not define a complete sequence, he can skip enough numbers to insure that the discontinuity degree of freedom remains independent. Since the analyst may choose to increase the joint number by any number from one to eight, he has considerable latitude in treating latitude in treating discontinuities.

Suppose it is desired to assign gridpoint continuity numbers to the plane framework shown in Figure 7-1. Here member c and d are hinged to joint 8.

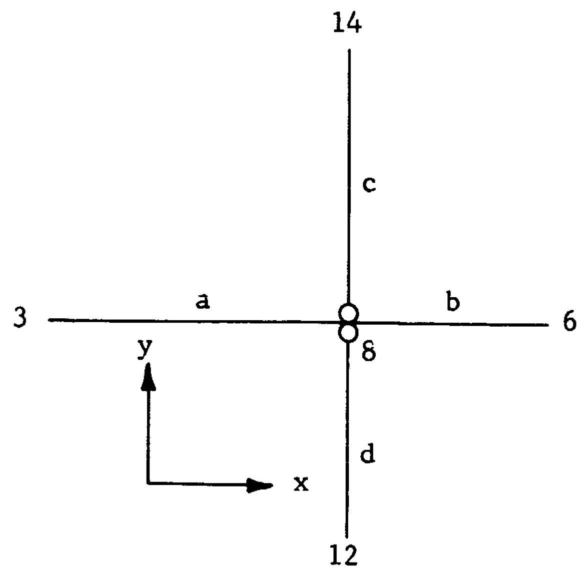



**Figure 7-1**

The continuity number for gridpoint 8 for members a and b could be all zeros, or left blank since both must be compatible with joint 8 displacements. The continuity number for member c at 8 could be 0000001. The continuity number for member d at 8 could be 0000002. The rotation in element c about the z axis (component 6) and at gridpoint 8 would be indicated by 96 in the output. The rotation in element d at 8 would be indicated by 106 in the output. Since joints 9 and 10 do not occur elsewhere in the problem, there is no ambiguity in the output.

If the deformation component is to be fixed (set at zero) at a joint, the pertinent digit is chosen as 9. Thus if a gridpoint is "built-in" the discontinuity number would be 999999. If it were to be pinned so that no displacements could occur, the discontinuity number would be 999000. Note that if all gridpoints meeting at a joint are fixed, then the joint is fixed in the same way. Thus, the analyst may impose all zero deformation boundary conditions as the equations are generated rather than in a separate WASH operation, if he chooses.

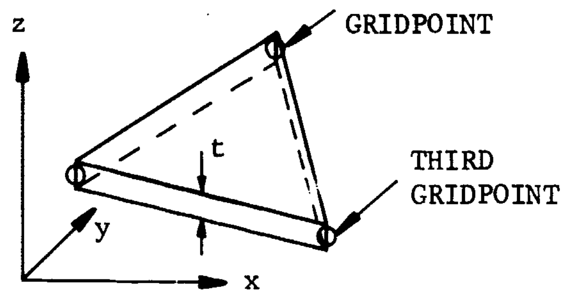
The organization of input data is shown in Tables 7-2 and 7-3. Data has been ordered to minimize the number of cards of input required to treat problems of most frequent occurrence. These cards are sufficient to define the gridpoint numbers, local geometry (thickness for the Facet, cross-sectional properties for the line element), material type, and gridpoint locations in the overall coordinate system. Including card two provides the data required to generate equations for analyses in acceleration, pressure, and thermal environments.

Cards are based on a seven column module called a field. The first field of each card is divided into three parts: the first part is a single column wide and contains the card number, the second part is three columns wide to define the number of the element, the third part is three columns wide and contains a blank and the identification used for the element assumptions. On all but the first card, the element identification may be left blank, if desired. Similarly, element number may be omitted on all but the first card. However, it is desirable to include it on all cards so that the cards can be automatically sorted if they get out of sort. The remainder of each card contains nine fields of floating decimal input and an alphanumeric field. Only the first two columns of the alphanumeric field are used in the program. The analyst can, however, use the full field, if he desires.

A detailed description of input data for the Facet and line elements is given in Tables 7-5 and 7-6. Units used in input are pounds, inches, seconds, and degrees Rankine.

TABLE 7-5

FACET ELEMENT DATA




| Card | Columns | Format | Item* | Interpretation |
|---|---|---|---|---|
| 1 | 1 | I1 | 1 | Indicates card number = 1. |
| 1 | 2-4 | I3 | K | Element number. $0 < K \leq 999$. |
| 1 | 5-7 | 1X, I2 | 31 | Specifies the Facet element assumptions. The first digit, 3, indicates three gridpoints are required. |
| 1 | 8-14 | E7.0 | GA | First, second, and third elastic gridpoint numbers for the element. If negative, solution displacements are produced in the local coordinate system at the gridpoint. |
| | 15-21 | E7.0 | GB | |
| | 22-28 | E7.0 | GC | |
| 1 | 29-35 | E7.0 | - | Data in this field will be ignored. |
| 1 | 36-42 | E7.0 | SA | First, second, and third substitute gridpoint numbers. If the gridpoint numbers are negative, solution displacements are produced in the local coordinate system at the gridpoint. |
| | 43-49 | | SB | |
| | 50-56 | | SC | |
| 1 | 57-63 | E7.0 | M | Facet mass.<br>$M \leq 0$, total mass: $\#sec^2/in$<br>$M \geq 0$, mass per unit area: $\#sec^2/in^3$ |
| 1 | 64-70 | E7.0 | T | Facet thickness in inches. |
| 1 | 71-72 | A2 | N | Name of the structural material. The first two characters of the name must match the first two characters of the material name in the material table. |

* Numbers given in this column are to be taken literally. The user must substitute appropriate numbers for letters given.

TABLE 7-5 (Cont'd)

FACET ELEMENT DATA

| Card | Columns | Format | Item* | Interpretation |
|---|---|---|---|---|
| 2 | 1 | I1 | 2 | Indicates card number = 2. |
| 2 | 2-4 | I3 | K | Element number. $0 \leqslant K \leqslant 999$. (may be omitted) |
| 2 | 5-7 | 1X,I2 | 31 | Same as Card 1, columns 5-7. (may be omitted) |
| 2 | 8-14 | E7.0 | CA | Continuity boundary conditions at the first second and third grid- |
| | 15-21 | E7.0 | CB | points. If both substitute points are given, these apply to the sub- |
| | 22-28 | E7.0 | CC | stitute points. If only elastic appear, they apply to the elastic. |
| 2 | 29-35 | E7.0 | - | Ignored |
| 2 | 36-42 | E7.0 | p | Normal pressure: pounds/inch$^2$ Positive in the plus z direction. |
| 2 | 43-49 | E7.0 | $T_m$ | Temperature (Degrees Rankine) of the material (used to define elastic constants). |
| 2 | 50-56 | E7.0 | $T_u$ | Upper surface temperature change (Degrees Rankine) from zero stress temperature. |
| 2 | 57-63 | E7.0 | $T_1$ | Lower surface temperature change (Degrees Rankine) from zero stress temperature. |
| 2 | 64-70 | E7.0 | - | Ignored |
| 2 | 71-72 | A2 | - | Ignored |
| 3 | Omit | | | |
| 4 | Omit | | | |

* Numbers given in this column are to be taken literally. The user must substitute appropriate numbers for letters given.

TABLE 7-5 (Cont'd)

FACET ELEMENT DATA

| Card | Columns | Format | Item* | Interpretation |
|---|---|---|---|---|
| 5 | 1 | I1 | 5 | Indicates card number = 5. |
| 5 | 2-4 | I3 | K | Element number. $0 \leqslant K \leqslant 999$. (may be omitted) |
| 5 | 5-7 | 1X,I2 | 31 | Same as Card 1, columns 5-7. (may be omitted) |
| 5 | 8-70 | 9E7.0 | | Coordinates (inches) in the overall system of the origin of the local coordinate system $(x_1,y_1,z_1)$; a point on the local x axis $(x_2,y_2,z_2)$; and a point in the x-y plane $(x_3,y_3,z_3)$, noncollinear with the first two points are selected to define desired direction of the local z axis. |
| 5 | 71-72 | A2 | - | Ignored |
| 6 | Omit | | | |
| 7 | 1 | I1 | 7 | Indicates card number = 7. |
| 7 | 2-4 | I3 | K | Element number. $0 \leqslant K \leqslant 999$. (may be omitted) |
| 7 | 5-7 | 1X,I2 | 31 | Same as Card 1, columns 5-7 (may be omitted) |
| 7 | 8-19 | 9E7.0 | | Coordinates (inches) of the substitute gridpoints corresponding to the first, second and third elastic gridpoints. |
| 7 | 71-72 | A2 | C | Coordinate identification. If C = L, substitute gridpoint coordinates are in the local system. If C ≠ L, coordinates are overall. |
| 8 | Omit | | | |
| 9 | 1 | I1 | 9 | Card number = 9. |
| 9 | 2-4 | I3 | K | Element number. $0 \leqslant K \leqslant 999$. (may be omitted) |

* Numbers given in this column are to be taken literally. The user must substitute appropriate numbers for letters given.

TABLE 7-5 (Cont'd)

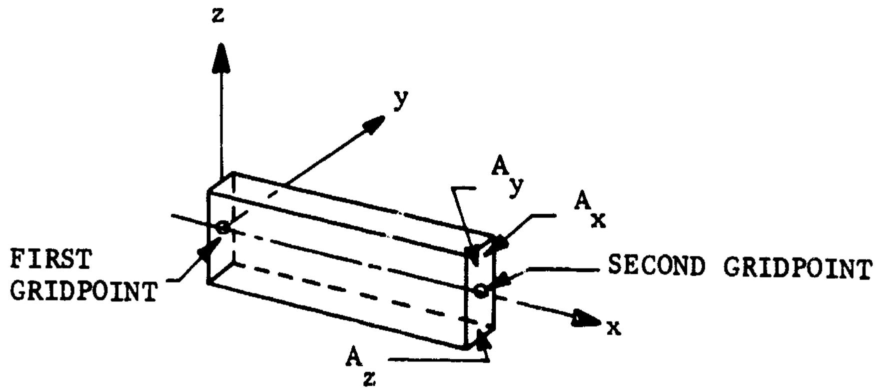
FACET ELEMENT DATA

| Card | Columns | Format | Item* | Interpretation |
|---|---|---|---|---|
| 9 | 5-7 | 1X,I2 | 31 | Same as Card 1, columns 5-7. (may be omitted) |
| 9 | 8-70 | 9E7.0 | | Coordinates (inches) for the first $(x_1, y_1, z_1)$, second $(x_2, y_2, z_2)$ and third $(x_3, y_3, z_3)$ elastic gridpoints of the Facet in the overall or local system. |
| 9 | 71-72 | A2 | C | Coordinate identification. If C = L elastic gridpoint coordinates are in the local system. If C ≠ L, coordinates are overall. |

* Numbers given in this column are to be taken literally. The user must substitute appropriate numbers for letters given.

## TABLE 7-6

## LINE ELEMENT DATA




| Card | Columns | Format | Item* | Interpretation |
|---|---|---|---|---|
| 1 | 1 | I1 | 1 | Indicates card number = 1. |
| 1 | 2-4 | I3 | K | Element number. $0 < K \leqslant 999$. |
| 1 | 5-7 | 1X,I2 | 2A | Specifies the line element equations. Rod and tube equations and (A = 1) elementary beam or (A = 2) shear beam displacement assumptions. |
| 1 | 8-14 | E7.0 | GA | First, second, and third gridpoint numbers for the element. The grid |
| | 15-21 | E7.0 | GB | number of the third gridpoint may be omitted. If numbers are negative, |
| | 22-28 | E7.0 | GC | solution displacements are in the local coordinate system at the gridpoint. |
| 1 | 29-35 | E7.0 | $A_x$ | Crossectional area (axial) in the y-z plane: inches$^2$. |
| 1 | 36-42 | E7.0 | $J_x$ | Torsional rigidity against twist about the x axis: inches$^4$. |
| 1 | 43-49 | E7.0 | $A_y$ | Effective area deforming in shear in the x-y plane due to a y force: inches$^2$. |
| 1 | 50-56 | E7.0 | $I_z$ | Moment of inertia resisting a moment about the z axis: inches$^4$. |

* Numbers given in this column are to be taken literally. The user must substitute appropriate numbers for letters.

TABLE 7-6 (Cont'd)

LINE ELEMENT DATA

| Card | Columns | Format | Item* | Interpretation |
|---|---|---|---|---|
| 1 | 57-63 | E7.0 | $A_z$ | Effective shear area for a shear force in the z direction: inches$^2$. |
| 1 | 64-70 | E7.0 | $I_y$ | Moment of inertia resisting a twist about the y axis: inches$^4$. |
| 1 | 71-72 | A2 | N | Name of the structural material. The first two characters of the name must match the first two characters of the material table material name. |
| 2 | 1 | I1 | 2 | Card number = 2. |
| 2 | 2-4 | I3 | K | Element number. $0 < K < 999$. |
| 2 | 5-7 | 1X,I2 | 2A | Same as card 1, columns 5-7. (may be omitted) |
| 2 | 8-21 | 2E7.0 | CA CB | Continuity boundary conditions at the first and second gridpoints. If substitute gridpoints are given, these apply to the substitute points. If only elastic appear, they apply to the elastic. |
| 2 | 22-28 | E7.0 | - | Ignored |
| 2 | 29-35 | E7.0 | M | Line element mass. If $M \leqslant 0$, M is the total mass (#sec$^2$in). If $M \geqslant 0$, M is the mass per unit length (#sec$^2$/in$^2$). |
| 2 | 36-42 | E7.0 | p | Normal pressure: pounds/inch (positive in the plus z direction). |
| 2 | 43-49 | E7.0 | $T_m$ | Temperature (degrees Rankine) of the material (used to define elastic constants). |
| 2 | 50-56 | E7.0 | $T_o$ | Mean temperature change (degrees Rankine per unit of crossectional area - $A_x$) of the element from the zero stress temperature. |

* Numbers given in this column are to be taken literally. The user must substitute appropriate numbers for letters.

TABLE 7-6 (Cont'd)

LINE ELEMENT DATA

| Card | Columns | Format | Item* | Interpretation |
|---|---|---|---|---|
| 2 | 57-63 | E7.0 | $T_y$ | Temperature gradient (degrees Rankine per unit crossectional moment of inertia $I_y$) in the z direction. |
| 2 | 64-70 | E7.0 | $T_z$ | Temperature gradient (degrees Rankine per unit crossectional moment of inertia $I_y$) in the y direction. |
| 2 | 71-72 | A2 | - | Ignored |
| 3 | Omit | | | |
| 4 | Omit | | | |
| 5 | 1 | I1 | 5 | Indicates card number = 5. |
| 5 | 2-4 | I3 | K | Element number. $0 \leqslant K \leqslant 999$. (may be omitted) |
| 5 | 5-7 | 1X,I2 | 2A | Same as Card 1, columns 5-7. (may be omitted) |
| 5 | 8-70 | 9E7.0 | - | Coordinates (inches) in the overall system of the origin of the local coordinate system, $(x_1, y_1, z_1)$; a point on the local x axis $(x_2, y_2, z_2)$; and a point in the x-y plane $(x_3, y_3, z_3)$, noncollinear with the first two points and located to define desired plus z direction. |
| 5 | 71-72 | A2 | - | Ignored |
| 6 | Omit | | | |
| 7 | 1 | I1 | 7 | Card number = 7. |
| 7 | 2-4 | I3 | K | Element number. $0 \leqslant K \leqslant 999$. |
| 7 | 5-7 | 1X,I2 | 2A | Same as Card 1, columns 5-7. (may be omitted) |

* Numbers given in this column are to be taken literally. The user must substitute appropriate numbers for letters.

TABLE 7-6 (Cont'd)

LINE ELEMENT DATA

| Card | Columns | Format | Item* | Interpretation |
|---|---|---|---|---|
| 7 | 8-21 | 2E7.0 | SA SB | First and second substitute gridpoint numbers. If the gridpoint numbers are negative, solution displacements are produced in the local coordinate system at the gridpoint. |
| 7 | 22-28 | E7.0 | - | Ignored |
| 7 | 29-70 | 6E7.0 | - | Coordinates (inches) of the substitute gridpoints corresponding to the first and second elastic gridpoints: all in the local or overall system. |
| 7 | 71-72 | A2 | C | Coordinate identity. If C = L, substitute gridpoint coordinates are in the local system. If C ≠ L, coordinates are overall. |
| 8 | Omit | | | |
| 9 | 1 | I9 | 9 | Card number = 9. |
| 9 | 2-4 | I3 | K | Element number. $0 \leq K \leq 999$. (may be omitted) |
| 9 | 5-7 | 1X,I2 | 2A | Same as Card 1, columns 5-7. (may be omitted) |
| 9 | 8-70 | 9E7.0 | - | Coordinates (inches) for the first $(x_1,y_1,z_1)$, and second $(x_2,y_2,z_2)$ elastic gridpoints of the line element and the gridpoint $(x_3,y_3,z_3)$ defining the principal plane of the crossection: all in the overall or local systems. |
| 9 | 71-72 | A2 | C | Coordinate identification. If C = L, elastic gridpoint coordinates are in the local system. If C ≠ L, coordinates are in the overall system. |

* Numbers given in this column are to be taken literally. The user must substitute appropriate numbers for letters.

A sample set of element input is shown in Table 7-7 for a structure composed of a single Facet bounded by a beam. Beam data is defined in cards 1 through 4; Facet on cards 5 through 9. The structure is clamped at gridpoint 1 (as noted in card 6, field 3). It is loaded with a uniform pressure of 13 pounds per inch along the beam (card 2, field 6) and 23 psi on the Facet (card 6, field 6) and the dead load (total weight of beam) of 6 pounds (card 2, field 5) and weight per unit area of .009 psi (card 5, field 9) for the Facet. The average temperature of the beam is 530$^{o}$R, with a mean temperature of 51$^{o}$R above the stress free state, and with a gradient of 11$^{o}$R/per unit of I in the y direction and 17$^{o}$R in the z (card 2, fields 7, 8, 9, and 10). The Facet has a mean temperature of 580$^{o}$R with the upper surface having a 30$^{o}$ decline (card 6, fields 7, 8, and 9). The beam has a 2 x .75 inch rectangular cross section. Assuming it is shear effectivity in tension, is .900*, its effective shear crossectional areas are 1.35 and 1.35 and the total area resisting axial force, 1.50(card 1, fields 8, 9, and 5). Since its torsional rigidity is .220**, the effective moments of inertia for moment vectors about the beams local x, y, and z axes is .220 in$^4$, .0703 in$^4$, and .50 in$^4$: card 1, fields 6, 10, and 8. (Note that the local z of the beam is in the plane of the Facet because of the choice of coordinates of the third beam gridpoint). The thickness of the Facet is .16 inches (card 5, field 10).

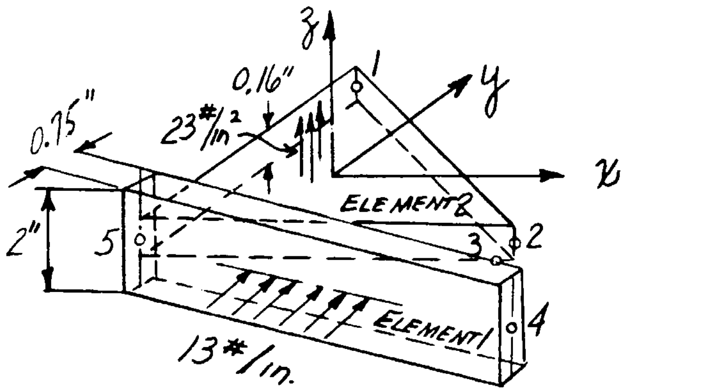
---

* Timoshenko, S., Theory of Elastic Stability, McGraw-Hill Book Co., New York, 1936, p.139.

** Timoshenko, S., Strength of Materials Part I, D. Van Nostrand Co., Inc., Princeton, N. J., 1955, p.290.

## TABLE 7-7

## SAMPLE ELEMENT DATA




1 @ (-3.0,6.0,0)
2 @ (0,12.0,0)
3 @ (12.0,-7.0,-.08)
4 @ (12.0,-7.0,-1.08)
5 @ (-12.0,-7.0,0)

Material = 2014T6 aluminum .009#/sq.in. of .75 in. thick.

| FIELD | 1 | | 2 | 3 | | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FORMAT | I1 | I3 | 1X,I2 | E7.0 | E7.0 | E7.0 | E7.0 | E7.0 | E7.0 | E7.0 | E7.0 | E7.0 | A2,4X |
| CARD 1 | 1 | 1 | 21 | 5 | 4 | 3 | 1.5 | .220 | 1.35 | .0703 | 1.35 | .50 | 2014T6 |
| CARD 2 | 2 | 1 | 21 | 0 | 0 | --- | -6 | 13 | 530 | 51 | 11 | 17 | --- |
| CARD 3 | 7 | 1 | --- | --- | 3 | --- | --- | --- | --- | 12.0 | -7.0 | -.08 | -G |
| CARD 4 | 9 | 1 | --- | -12.0 | -7.0 | 0 | 12.0 | -7.0 | -1.08 | 0 | 12.0 | -7.0 | G |
| CARD 5 | 1 | 2 | 31 | 1 | 5 | 2 | --- | --- | --- | 3 | .009 | .16 | 2014T6 |
| CARD 6 | 2 | 2 | --- | 999999 | 0 | 0 | --- | 23.0 | 580 | 30 | -30 | --- | --- |
| CARD 7 | 5 | 2 | --- | -12.0 | -7.0 | 0 | 12.0 | -7.0 | 0 | 0 | 0 | 2.6 | --- |
| CARD 8 | 7 | 2 | --- | --- | --- | --- | --- | --- | --- | -12.0 | -7.0 | 0 | G |
| CARD 9 | 9 | 2 | --- | 9.0 | 13.0 | 0 | 0 | 0 | 0 | 24.0 | 0 | -1.08 | L |

Coordinates of all gridpoints for the beam are given in the overall coordinate system (field 11 on cards 3 and 4). The elastic line runs from gridpoint 5 to 4 (card 4, fields 1 through 10). Transfer of loads to the beam occurs at the substitute gridpoint 3 at the 4 end of the beam as required on card 3, fields 1-10. All continuity conditions are zero, so there are no restraints to be imposed on the beam at the element level.

Coordinates of all elastic gridpoints for the Facet are in a local system and substitutes are in the overall (noted in field 11, cards 9 and 8). Coordinates of the local system origin given in the overall system are (-12.0, -7.0, 0.0), point on the local x axis is at (12.0, -7.0, 0) and another point in the local x-y plane is (0.0, 2.6, 0.0) as given on card 7, fields 2 through 10. This defines the local axis to be along side 5-2 of the Facet and the x-y plane to coincide with the Facet elastic plane. Coordinates of the three gridpoints defining the elastic plane and apexes of the Facet, in the local coordinate system are (-7.0, 5.0, 0.0), (24.0, 0.0, 0.0), and (0.0, 0.0, 0.0) as given on card 9, fields 2 through 10. Load transfer for gridpoint 2 occurs at the substitute gridpoint 3 (card 5, field 8).

Both the elementary beam and the Facet are fabricated of 2014T6 aluminum (cards 1 and 5, field 11).

This problem provides an idea of the flexibility of introducing input and the complexity of the structural problems that can be represented. In a practical problem, input data is usually more error free if data is prepared in groups rather than by element. Thus all cards are prepared together, all overall coordinate data together, all local coordinate data together and all geometric data together. This permits the use of ditto marks rather than data transcription to indicate duplication to the key-puncher. It permits the analyst to scan like quantities and locate and evict anomalies.

## 7.4 MATRIX DATA

Matrix data format and interpretation are defined in Section 6, usage writeups in the operation subprogram READ. Points of particular importance, i.e., formats and sort, are considered here.

There are two permissible formats, coded and precoded. It matrix data is not in one of these formats a card reading stop will usually occur. For a detailed description of coded and precoded formats see Section 2-4.

Matrix data must conform with sort requirements. The identification card must be the first card. If the matrix is in coded format the sort must correspond to the sort designated on the identification card. The number of cards must be the same as defined on the identification card. If the matrix is in precoded format it must be in the order defined on the identification card and all zero terms must be included. The codes must appear first followed by the matrix elements.

Table 7-8 provides an example of matrix input data for the two types of formats. In addition to the column listed sort selected for the coded form, the matrix could be input in row listed form or unsorted. If unsorted, it would usually be sorted by the SORT link since almost all links require either row or column sorted matrices as input.

A detailed explanation of input interpretation is given with the READ link writeup in Section 6. The example in Table 7-8 illustrates some of the input features. The first card of the precoded format names the matrix MAT001, indicates that there are two cards of input to be read, the matrix is a 2x3, is row listed, and precoded. The second card contains the column codes followed by the row cards. A new card is started for the matrix elements which are given here in row sort. Note that all elements of the matrix are given explicitly.

The first card in the coded format is also an identification card. It names the matrix MAT001, indicates two cards of input, need not specify matrix order, defines input as column listed and coded. The second and third cards define the codes and corresponding elements of the matrix in column listing sequence. Note that the null element (13, 3) has been omitted in the input. It could have been included, if desired, but it is an implied zero if omitted anyway.

## 7.5 DECK ARRANGEMENT

A problem deck is made up of at most four components. These components are initiating deck, pseudo instructions, descriptive data, and matrix data and heading cards.

The initiating deck is invariant. It is a self-contained preprogrammed package that starts the SAMIS program. It must be made a part of every new problem and it must be the first component in a problem deck. It includes a chain link calling MAKER, the *DATA control card and the cards defining the table of link designations. These cards are stored as the ABA and ABB tables and are described in paragraph 4.2.

TABLE 7-8

SAMPLE MATRIX DATA

$$\text{MAT001} = \begin{array}{cc} & \begin{array}{ccc} 1 & 3 & 5 \end{array} \\ \begin{array}{c} 7 \\ 13 \end{array} & \begin{bmatrix} 11. & 12. & 13. \\ 21. & 0. & 23. \end{bmatrix} \end{array}$$

PRECODED INPUT FORM

| | | | | | | |
|---|---|---|---|---|---|---|
| FORMAT | (A3,I3) | I6 | I6 | I6 | I6 | I6 |
| CARD 1 | MAT001 | 2 | 2 | 3 | -1 | 1 |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FORMAT | I6 | I6 | I6 | I6 | I6 | I6 | I6 | I6 | I6 | I6 | I6 | I6 |
| CARD 2 | 1 | 3 | 5 | 7 | 13 | - | - | - | - | - | - | - |

| | | | | | | |
|---|---|---|---|---|---|---|
| FORMAT | E12.0 | E12.0 | E12.0 | E12.0 | E12.0 | E12.0 |
| CARD 3 | 11. | 12. | 13. | 21. | 0 | 23. |

CODED INPUT FORM

| | | | | | | |
|---|---|---|---|---|---|---|
| FORMAT | (A3,I3) | I6 | I6 | I6 | I6 | I6 |
| CARD 1 | MAT001 | 2 | 0 | 0 | 1 | 0 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| FORMAT | I6 | I6 | E12.0 | I6 | I6 | E12.0 | I6 | I6 | E12.0 |
| CARD 2 | 7 | 1 | 11. | 13 | 1 | 21. | 7 | 3 | 12. |
| CARD 3 | 7 | 5 | 13. | 13 | 5 | 23. | | | |

The pseudo instructions constitute the second component of a problem deck. They must be made up according to the rules and formats outline in paragraph 4.1. The last pseudo instruction must be a HALT.

Descriptive data is the third component. Descriptive data includes the "material tables" and the "element data" in that order. The material table must have at least three cards, two data cards and one zero stop card. The parameters of the cards and the associated formats are given in detail in paragraph 7.2. The element data follows the material table in the problem deck. The parameters and the formats are given in paragraph 7.3. The technical document for the BILD subroutine group used should also be consulted for pertinent theoretical details.

The fourth component is matrix information. Matrix information consists of matrix data and heading cards. Matrix data is described in paragraph 7.4 and in the READ link writeup in Section 6.

The first matrix data card must be the identification card. The heading cards can contain any comments the analyst desires. They must be in the same order as they are to be printed. There is no practical limit to the number of cards. These are described in the INKS writeup in Section 6.

In summary, deck arrangement is normally as follows:

Starter binary deck
*DATA card
ABA and ABB tables
Pseudo instructions
Material tables
Zeros card
Element data
Matrix data

# SECTION 8

# PROGRAM FLOW CHARTS

This section contains the SAMIS flow charts and storage maps. Charts are ordered to provide first a general picture of the interrelation between links and tapes then a definition of the relations between subroutines in a link, and finally the flow and mapping of critical subroutines. Flow charts are organized to provide an understanding of the sequences of operations and relate the sequence to the Fortran statements.

## 8.1 OVERALL FLOW CHART

Figure 8-1 depicts the flow of operations among the links of the SAMIS system. The calculations are initiated by the MAKER link, which prepares the problem pseudo program tape from card input. This link calls the MINTS link using the monitor subroutine CHAIN.

The MINTS link locates data on tape and in-core and selects the operation link required. After the operation link has completed its function, it returns control to MINTS using the subroutine CHAIN.

Figure 8-1 defines the tapes providing input to each link and those written as output. All tapes are written in binary except the problem pseudo program tape and the output print tape.

## 8.2 LINK FLOW CHARTS

Figures 8-2.1 through 8-2.4 depict flow charts for the MAKER, MINTS, BILD, and a typical operation link. These charts show tape usage during the performance of operations in the link. Link return is through the CHAIN routine. All subroutines required in normal execution of the link are indicated.

Table 8-1 shows the storage map for data which is accessible to all links. This table defines the function and size of each array in the SAMIS system COMMON. Included in the table is a list of all SAMIS links currently available. Note that these links do not correspond exactly with pseudo instructions. For example, the ADDS link performs either the ADDS or SUBS operation. Table 8-2 shows the arrays retained in COMMON especially for the BILD link. This map provides the interpretation of the arrays. The dummy variables and erasable arrays are used within particular subroutines to save storage space by reuse. The BILD map also includes a summary of subroutines required for this link. Note that mapping is dependent on the use of the IBM Fortran II version 3 compiler.

### 8.3 SUBROUTINE FLOW CHARTS

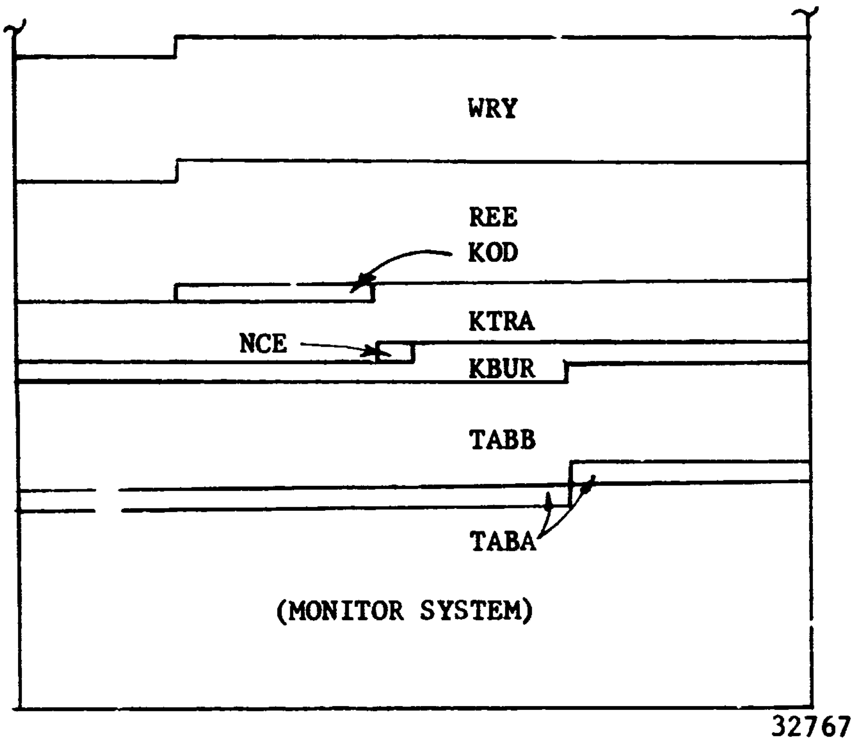
Detailed flow charts for all the MAKER, MINTS, and critical BILD subroutines are furnished in Figures 8-3, 8-4, and 8-5. Included with the MINTS subroutines are charts for all the "service" routines (e.g., TAPES). Critical subroutines in BILD are those used by more than one BILD subroutine group (i.e., input-output and transformation subroutines).

Within each box of the flow chart is a number contained in a rectangle. This number is coordinated with instruction numbers in the Fortran source deck. When the Fortran instruction has no number, the number in the box is augmented to indicate how many instructions before (minus) or after (plus) a numbered Fortran instruction the activity defined is initiated. When these numbers appear on the left hand side of a box, they indicate the instruction which calls the subroutine in question.

Capital and lower case letters are used to differentiate between link and subroutine terminals. When a subroutine starts or returns to another link subroutine, the start or return is noted with mixed upper and lower case letters. When the subroutine is the first routine of a link or returns through CHAIN to pick up the next link, the terminal is defined in upper case letters

**TABLE 8-1**

**SAMIS SYSTEM COMMON STORAGE MAP**




**SAMIS LINKS**

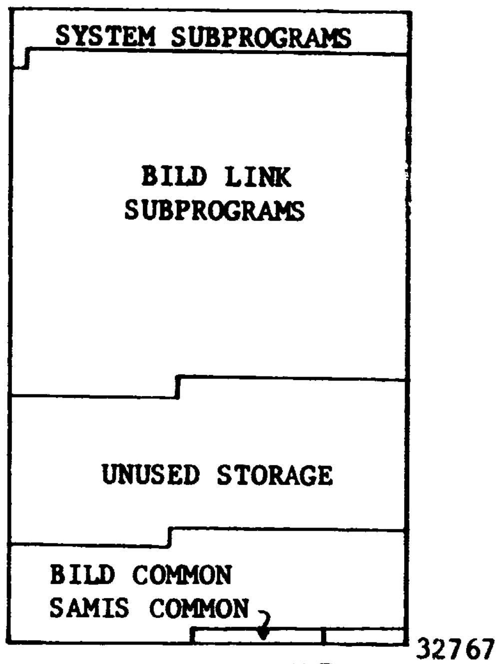
| Number | Name |
|---|---|
| 1 | READ |
| 2 | MAKER |
| 3 | COLS |
| 5 | FLIP |
| 6 | CODE |
| 7 | DECO |
| 8 | ADDS |
| 9 | MULT |
| 10 | WASH |
| 11 | ITER |
| 12 | CHIN |
| 13 | CHOL |
| 15 | ROOT |
| 16 | BILD |
| 18 | INKS |
| 26 | MINTS |

SAMIS COMMON

| Location | Item | Interpretation |
|---|---|---|
| 32229 | WRY(120) | Tape (output) buffer |
| 32349 | REE(120) | Tape (input) buffer |
| 32356 | KOD(5) | Array for code packing |
| 32416 | KTRA(60) | Matrix identification array |
| 32417 | NCE | Error Flag |
| 32441 | KBUR(24) | Current pseudo instruction |
| 32541 | TABB(100) | Table of link designations |
| 32561 | TABA(20) | Table of logic instructions |

## TABLE 8-2

## BILD LINK STORAGE MAP




GENERAL LAYOUT

**BILD SUBPROGRAMS**

| Label | Name | Label | Name |
|---|---|---|---|
| PO | --- | PN | ADD |
| PA | BILD | PØ | MUL |
| PB | FACET | PP | TABLE |
| PC | CØR | PQ | MØD |
| PD | TRN | PR | CØES |
| PE | LØD | PS | CØNE |
| PF | STF | PT | SPR |
| PG | STR | PU | TN |
| PH | SFT | PV | LD |
| PI | STFQ | PW | SF |
| PJ | TRS | PX | SR |
| PK | ADMS | PY | SFQ |
| PL | SAD | PZ | BEAM |
| PM | SCA | | |

**BILD COMMON**

| Location | Item | Interpretation |
|---|---|---|
| 25395 | Z31 | Z direction difference in overall coordinates |
| 25396 | Z21 | Z direction difference in overall coordinates |
| 25399 | Z1(3) | Overall Z coordinates of gridpoints |
| 25400 | YYY | Dummy |
| 25409 | YY(9) | 3 x 3 matrix of $Y_iY_j$; i = 1,2,3; j = 1,2,3 |
| 25418 | YX(9) | 3 x 3 matrix of $Y_iX_j$; i = 1,2,2; j = 1,2,3 |
| 25427 | YA(9) | Dummy |
| 25430 | Y(3) | Local Y coordinates of Facet joints |
| 25431 | Y32 | Y direction difference, overall coordinates |
| 25432 | Y31 | Y direction difference, overall coordinates |
| 25433 | Y21 | Y direction difference, overall coordinates |
| 25436 | Y4(3) | Dummy |
| 25439 | Y3(3) | Dummy |
| 25442 | Y2(3) | Local Y coordinates of gridpoints |
| 25445 | Y1(3) | Overall Y coordinates of gridpoints |
| 25446 | YO | Dummy |
| 25455 | XYYX(9) | Sum of XY and YX array |
| 25464 | XY(9) | 3 x 3 matrix of $X_iY_j$; i = 1,2,3; j = 1,2,3 |
| 25465 | XXX | Dummy |

BILD COMMON, Cont'd

| Location | Item | Interpretation |
|---|---|---|
| 25474 | XX(9) | 3 x 3 matrix of $X_iX_j$ i = 1,2,3; j = 1,2,3 |
| 25475 | XL | Length of line element |
| 25484 | XA | Dummy |
| 25487 | X | Local X coordinates of Facet joints |
| 25488 | X32 | X direction difference, overall coordinates |
| 25489 | X31 | X direction difference, overall coordinates |
| 25490 | X21 | X direction difference, overall coordinates |
| 25493 | X4(3) | Dummy |
| 25496 | X3 | Dummy |
| 25499 | X2 | Local X coordinates of gridpoints |
| 25502 | X1 | Overall X coordinates of gridpoints |
| 25503 | X0 | Dummy |
| 25504 | WT | Element mass input |
| 25505 | V3 | Direction cosines relating Facet surface to overall coordinate axes |
| 25506 | V2 | |
| 25507 | V1 | |
| 25508 | TYPE | Material name for the element |
| 25509 | TU | Facet upper surface temperature change |
| 25510 | TL | Facet lower surface temperature change |
| 25511 | TK | Temperature to define elastic constants |
| 25512 | THICK | Facet thickness |
| 25521 | T(3,3) | Erasable 3 x 3 array |
| 25522 | T6 | Parameters defined in subroutines as erasable data |
| 25523 | T5 | |
| 25524 | T4 | |
| 25525 | T3 | |
| 25526 | T2 | |
| 25527 | T1 | |
| 25528 | S | Dummies |
| 25529 | S312 | |
| 25530 | S232 | |
| 25531 | S231 | |
| 25532 | S223 | |
| 25533 | S212 | |
| 25534 | S122 | |
| 25535 | S31 | |
| 25536 | S23 | |
| 25537 | S1 | |
| 25538 | PRESS | Normal pressure intensity |
| 25539 | LOCAL | Local coordinate flags |
| 25587 | KRE(48) | Temporary storage of matrix codes |
| 25588 | J | Indexes and flags |
| 25589 | J3 | |
| 25590 | J2 | |
| 25591 | J1 | |
| 25592 | ITYPE | Element type |
| 25593 | ITEMP | Temperature option flag |

**BILD COMMON, Cont'd**

| Location | Item | Interpretation |
|---|---|---|
| 25594 | ISIZE | Stiffness matrix maximum order |
| 25595 | IPRES | Prestress flag |
| 25596 | IE | Element number |
| 25597 | I | |
| 25598 | I3 | Temporary indexes and flags |
| 25599 | I2 | |
| 25600 | I1 | |
| 25601 | H | Facet membrane stiffness scalar |
| 25602 | G | Facet bending stiffness scalar |
| 25737 | F1(15,9) | Loading matrix |
| 25738 | E3 | |
| 25739 | E2 | Direction cosines relating Facet surface to |
| 25740 | E1 | overall coordinate axes |
| 25741 | D1112 | |
| 25742 | D662 | |
| 25743 | D661 | |
| 25744 | D662 | |
| 25745 | D621 | |
| 25746 | D612 | |
| 25747 | D611 | |
| 25748 | D551 | |
| 25749 | D541 | Elastic constants modified to reflect plane |
| 25750 | D441 | stress condition |
| 25751 | D222 | |
| 25752 | D221 | |
| 25753 | D212 | |
| 25754 | D211 | |
| 25755 | D112 | |
| 25756 | D111 | |
| 25757 | D66 | |
| 25758 | D63 | |
| 25759 | D62 | |
| 25760 | D61 | |
| 25761 | D55 | |
| 25762 | D54 | |
| 25763 | D44 | Elastic constants interpolated from material |
| 25764 | D33 | tables |
| 25765 | D32 | |
| 25766 | D31 | |
| 25767 | D22 | |
| 25768 | D21 | |
| 25769 | D11 | |
| 25770 | D3 | Direction cosines relating Facet surface to |
| 25771 | D2 | overall coordinates |
| 25772 | D1 | |
| 25773 | C | Dummy |
| 25774 | C232 | Dummy |

BILD COMMON, Cont'd

| Location | Item | Interpretation |
|---|---|---|
| 25775 | C122 | Dummy |
| 25776 | C31 | Dummy |
| 25777 | C23 | Dummy |
| 25778 | C12 | Dummy |
| 25787 | C2 | Line direction cosines between input and local axes |
| 25796 | C1 | Line direction cosines between local and overall axes |
| 26012 | BT(36,6) | Transformation matrix - local to final coordinates |
| 26013 | AYY | Dummy |
| 26014 | AXY | Dummy |
| 26015 | AXX | Dummy |
| 26016 | AT | Coefficient of thermal expansion |
| 26017 | AREA | Planform area of Facet |
| 26018 | ALPHA | Dummy |
| 26027 | ALAL(9) | Erasable matrix storage |
| 26030 | ALA(3) | Erasable vector storage |
| 26033 | AL(3) | Erasable vector storage |
| 26258 | AK(15,15) | Stiffness matrix array |
| 26259 | A31 | Dummy |
| 26260 | A23 | Dummy |
| 26261 | A21 | Dummy |
| 26270 | A9(9) | |
| 26279 | A8(9) | |
| 26288 | A7(9) | |
| 26297 | A6(9) | |
| 26306 | A5(9) | 3 x 3 matrix components of the stiffness matrix |
| 26315 | A4(9) | |
| 26324 | A3(9) | |
| 26333 | A2(9) | |
| 26342 | A1(9) | |
| 26342 | W(3,3) | |
| 26425 | NATA(83) | Element input data |
| 27073 | EM(15,15) | Mass matrix array |
| 27298 | TR(15,15) | Erasable matrix storage |
| 27433 | S2(9,15) | Erasable matrix storage, stress matrix |
| 27505 | S1(8,9) | Erasable matrix storage |
| 28369 | S4(36,54) | Output array |
| 28909 | F(8,24) | Stress matrix |
| 28909 | S3(15,36) | Erasable matrix storage |

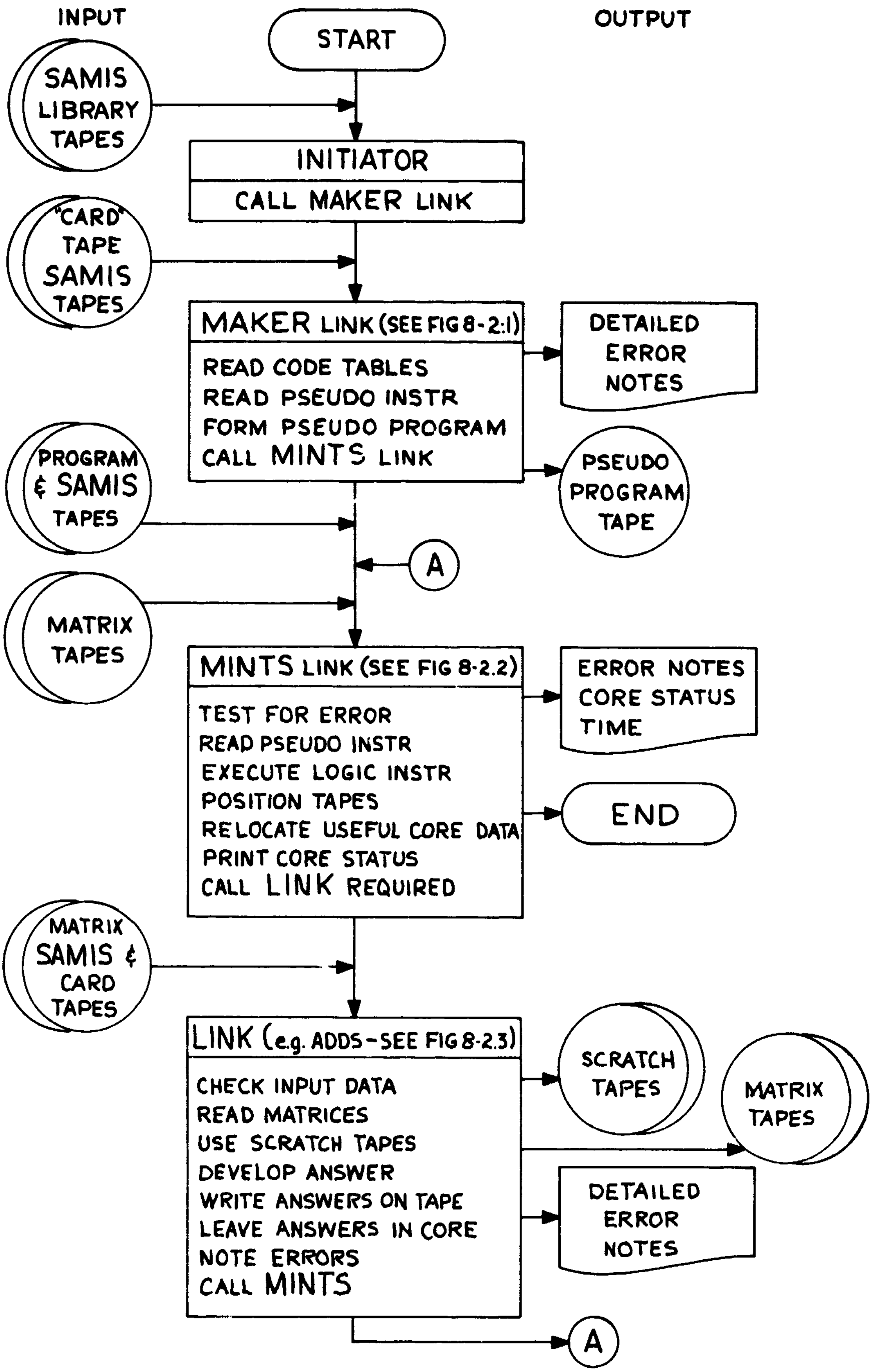



Figure 8-1 OVERALL FLOW CHART

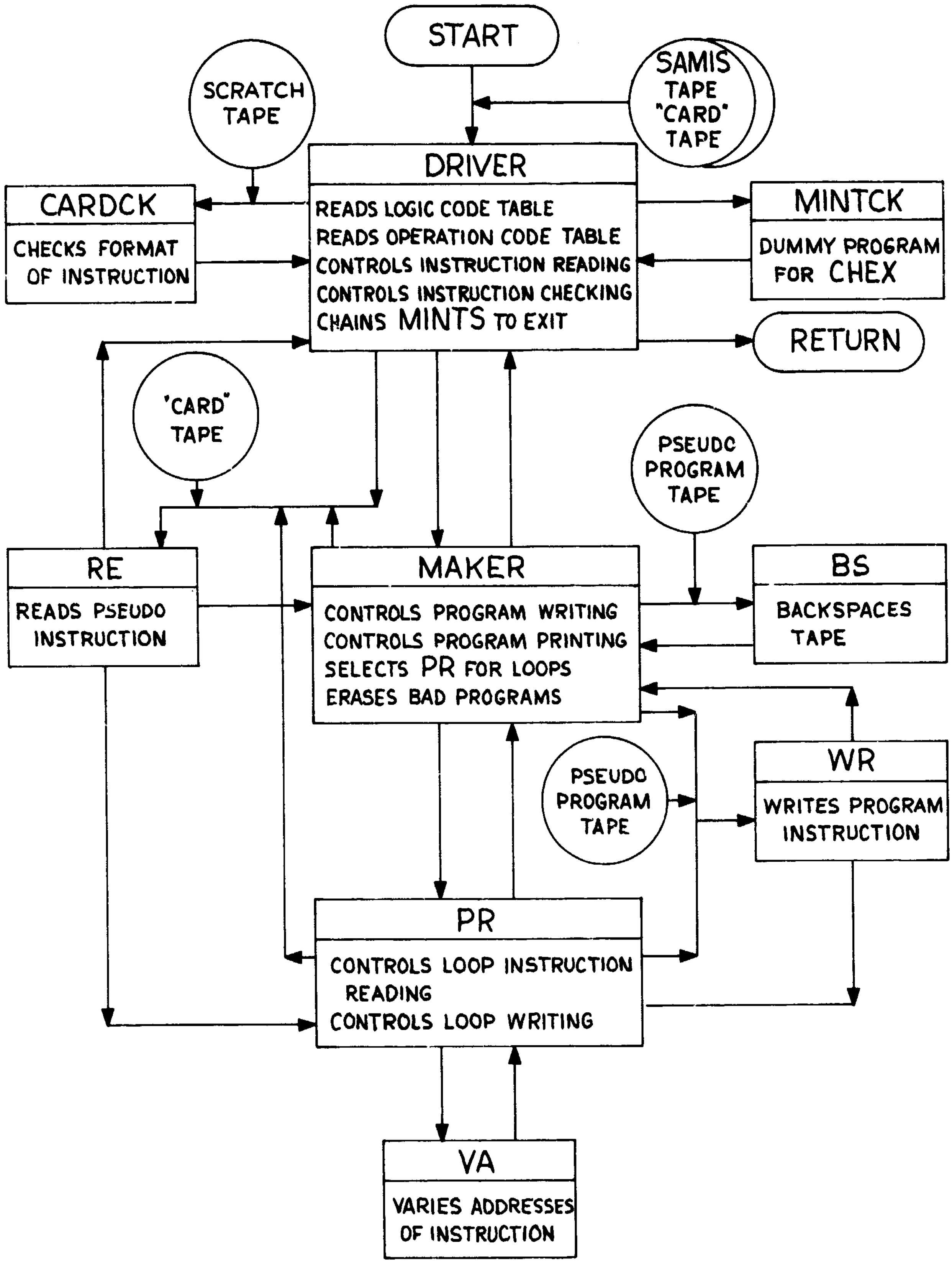



Figure 8-2.1 MAKER LINK FLOW CHART

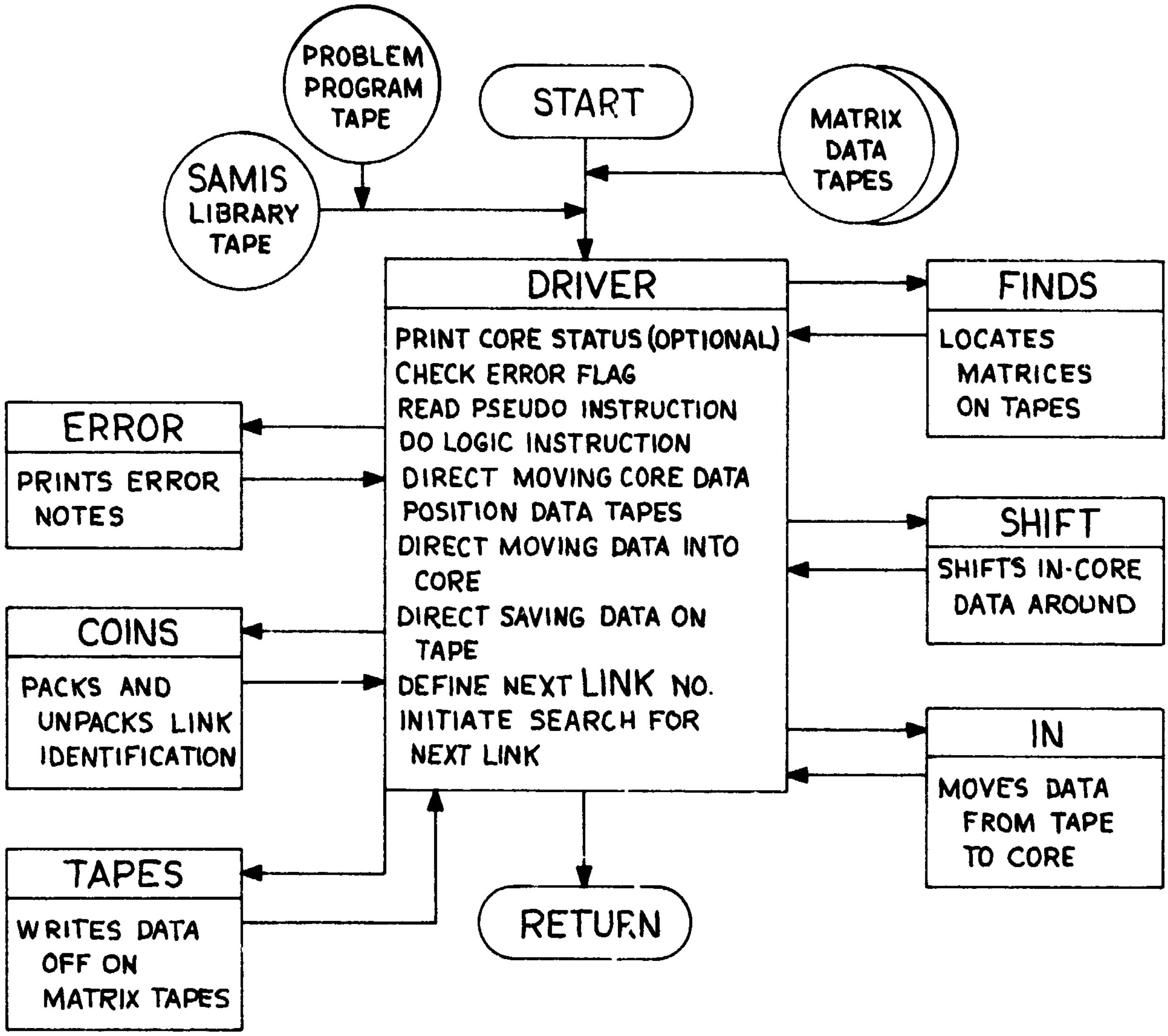



Figure 8-2.2 MINTS LINK FLOW CHART

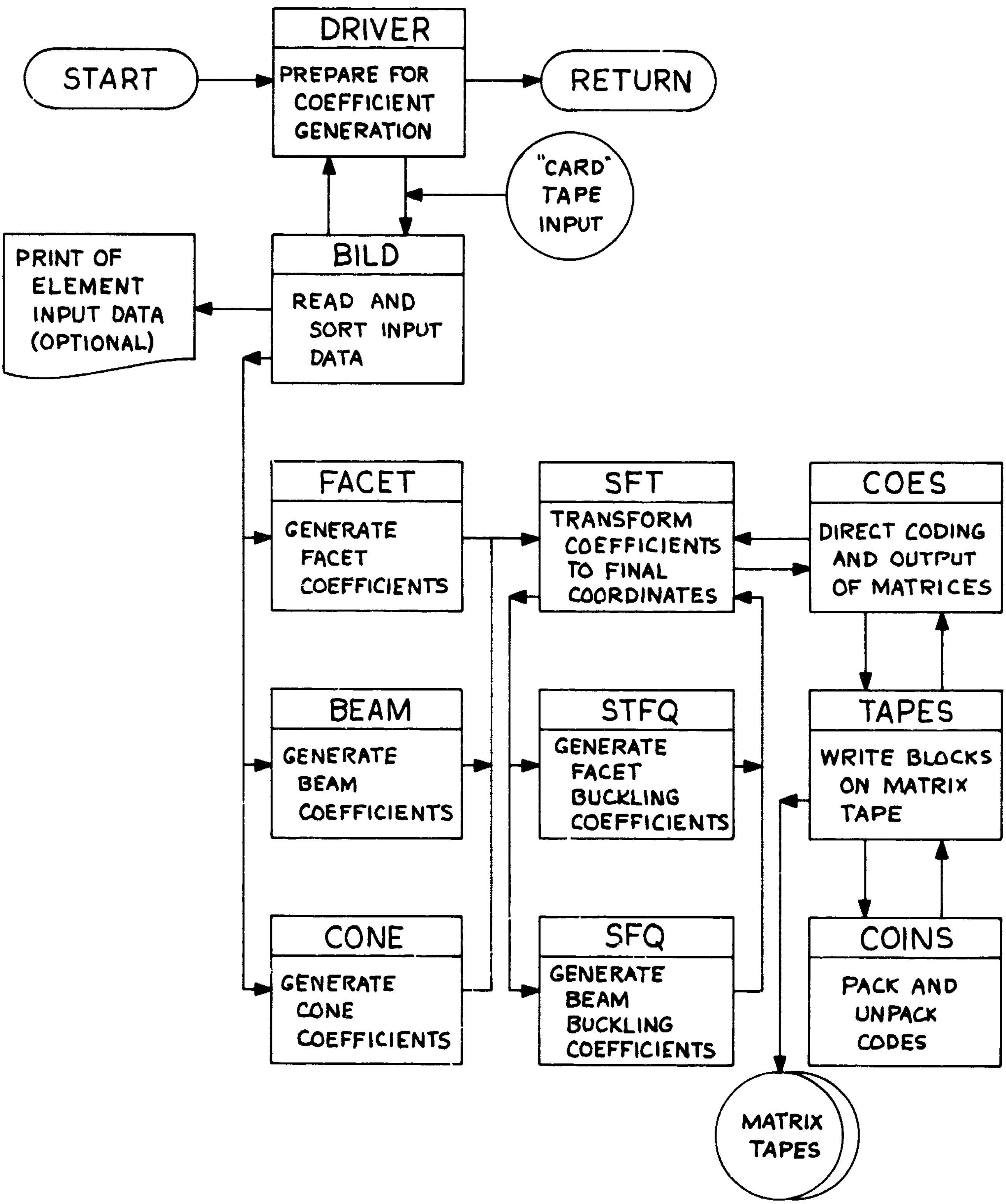



Figure 8-2.3 BILD LINK FLOW CHART

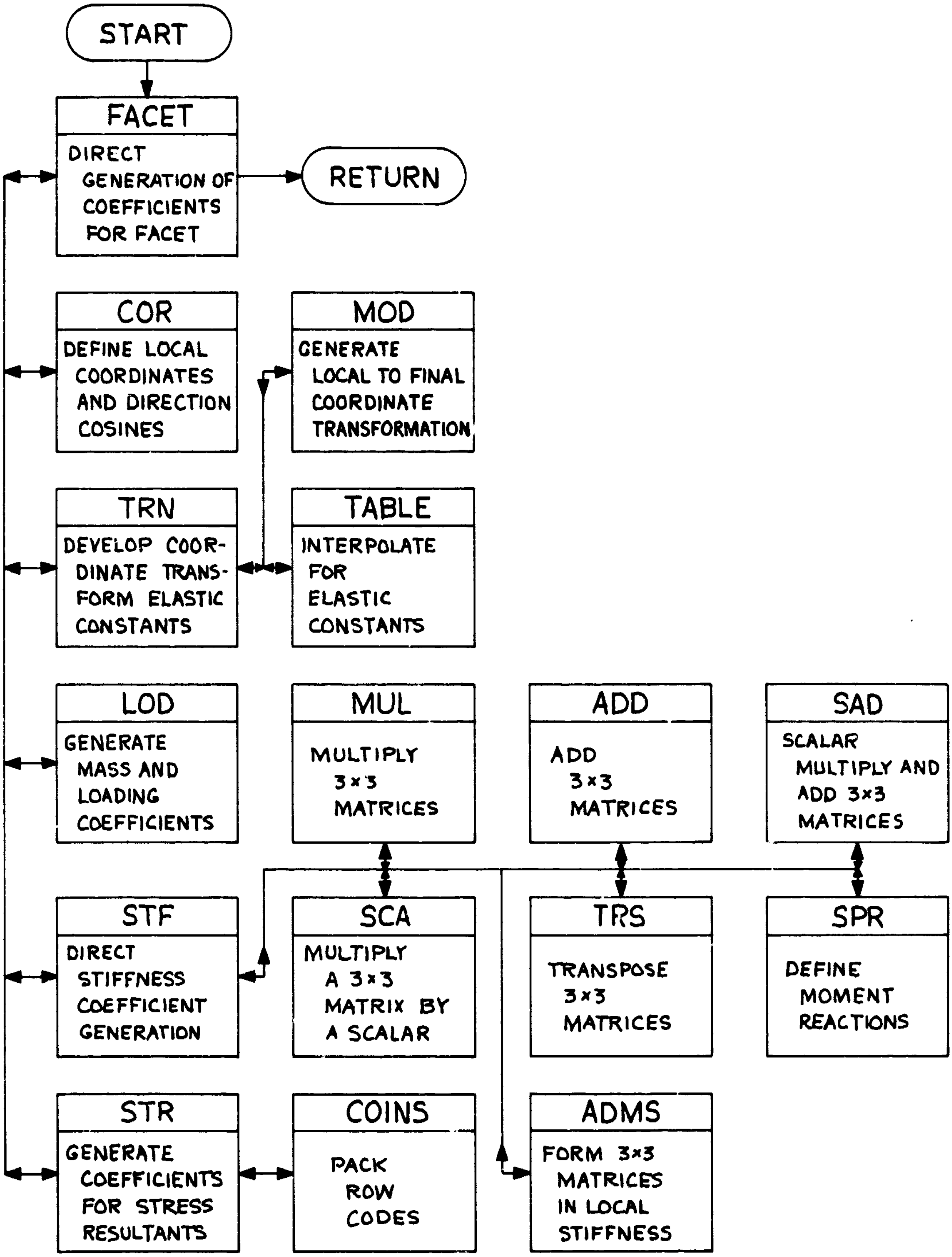



Figure 8-2.3 BILD LINK FLOW FACET SUBROUTINE GROUP

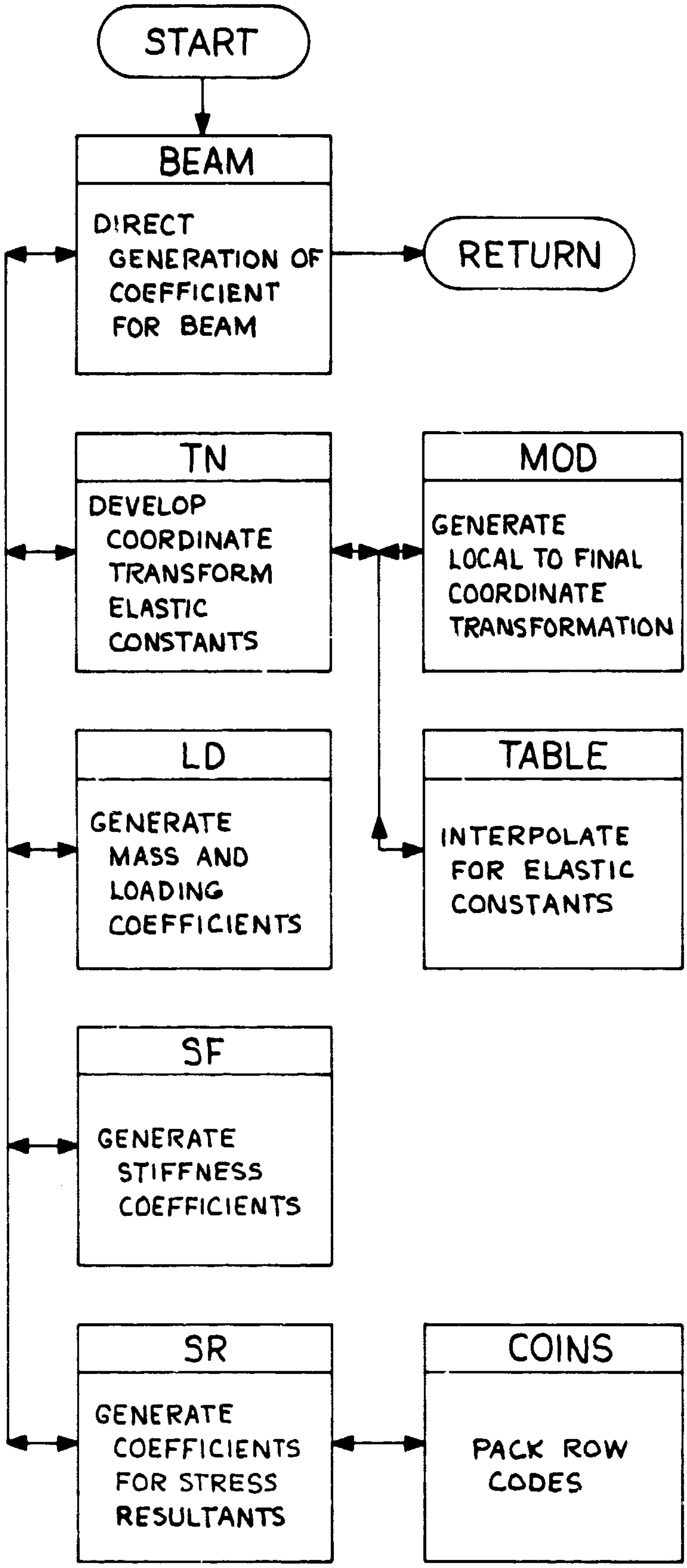



Figure 8-2.3 BILD LINK FLOW BEAM SUBROUTINE GROUP

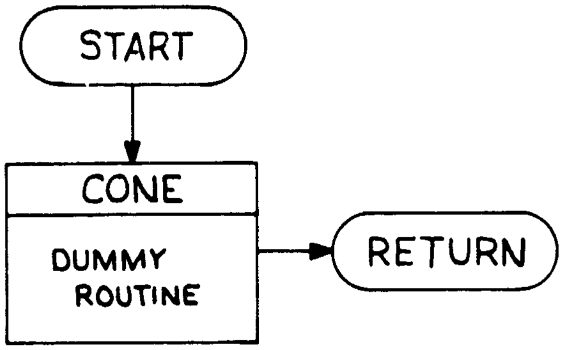



Figure 8-2.3 BILD LINK FLOW CONE SUBROUTINE GROUP

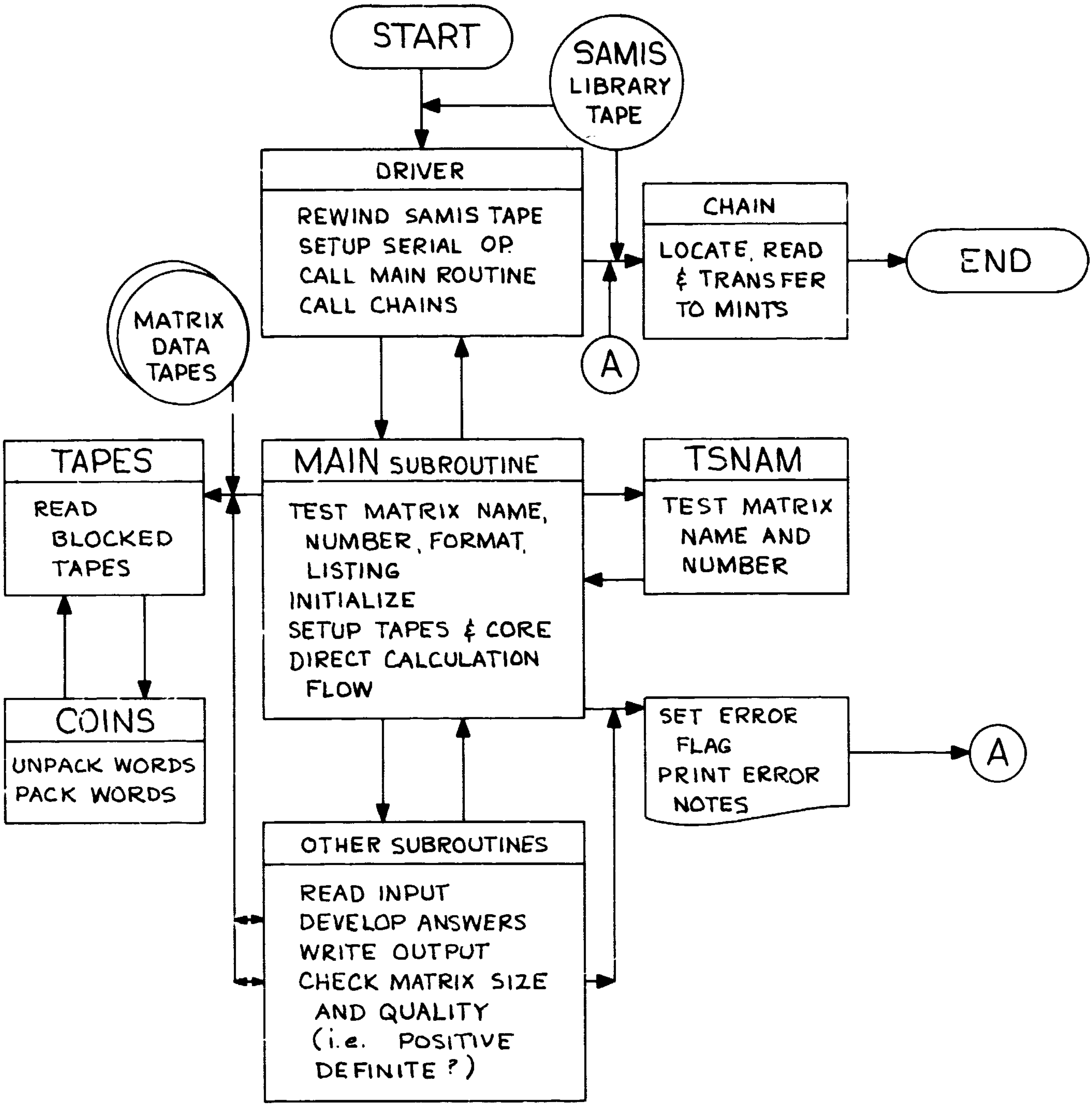



Figure 8-2.4 TYPICAL LINK FLOW CHART

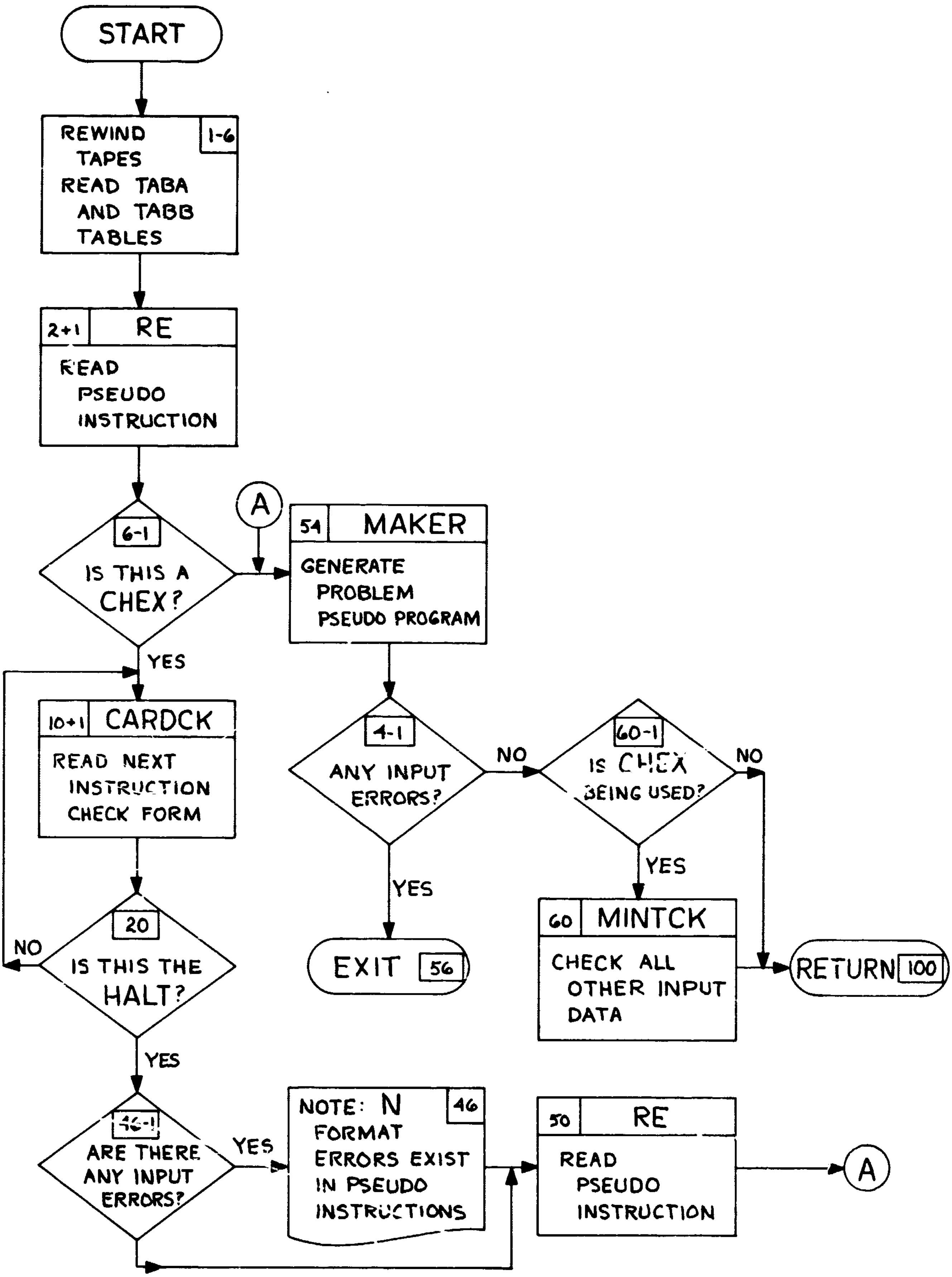



Figure 8-3.1 MAKER DRIVER FLOW

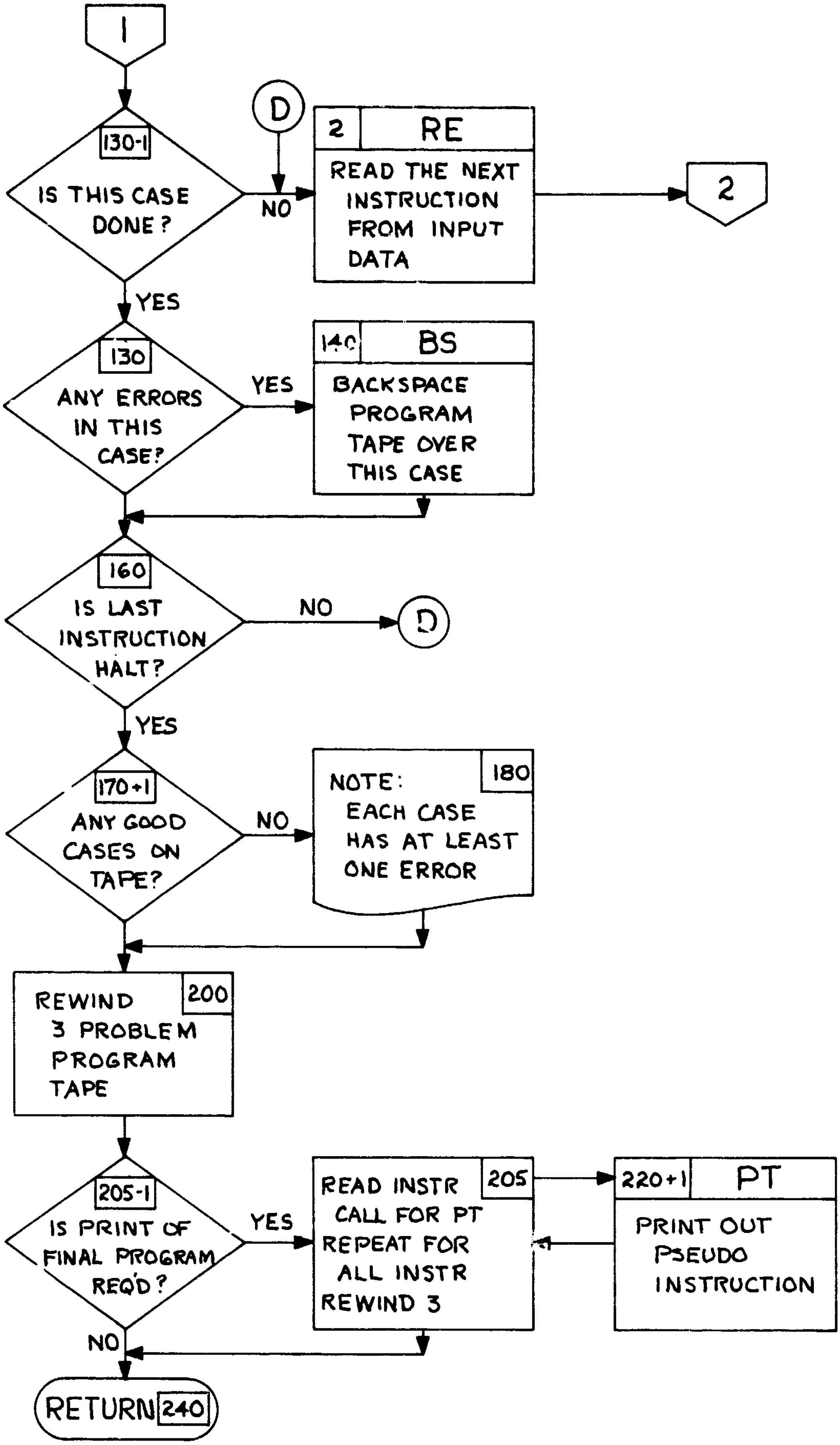



Figure 8-3.2 "MAKER" FLOW

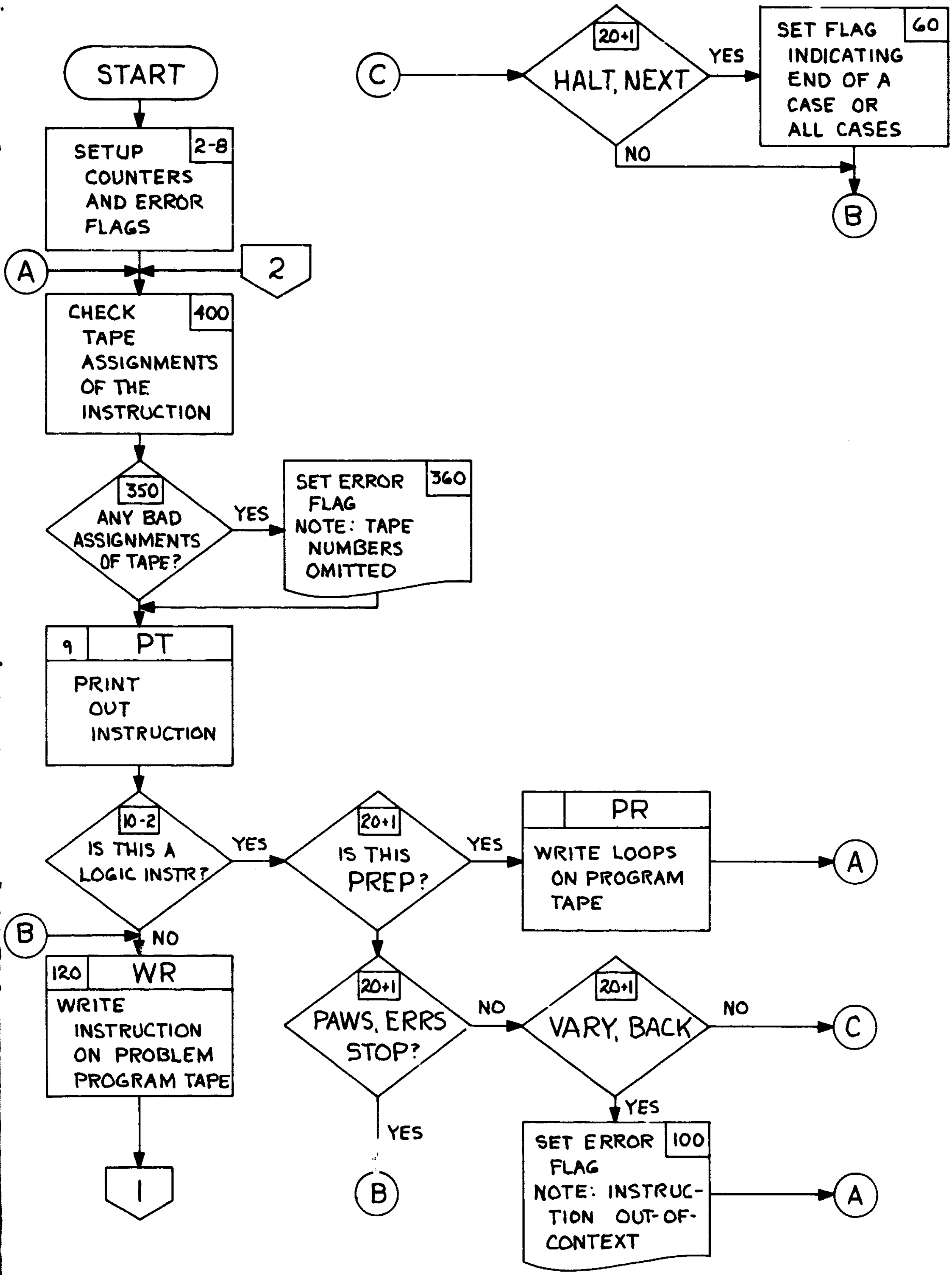



Figure 8-3.2 "MAKER" FLOW

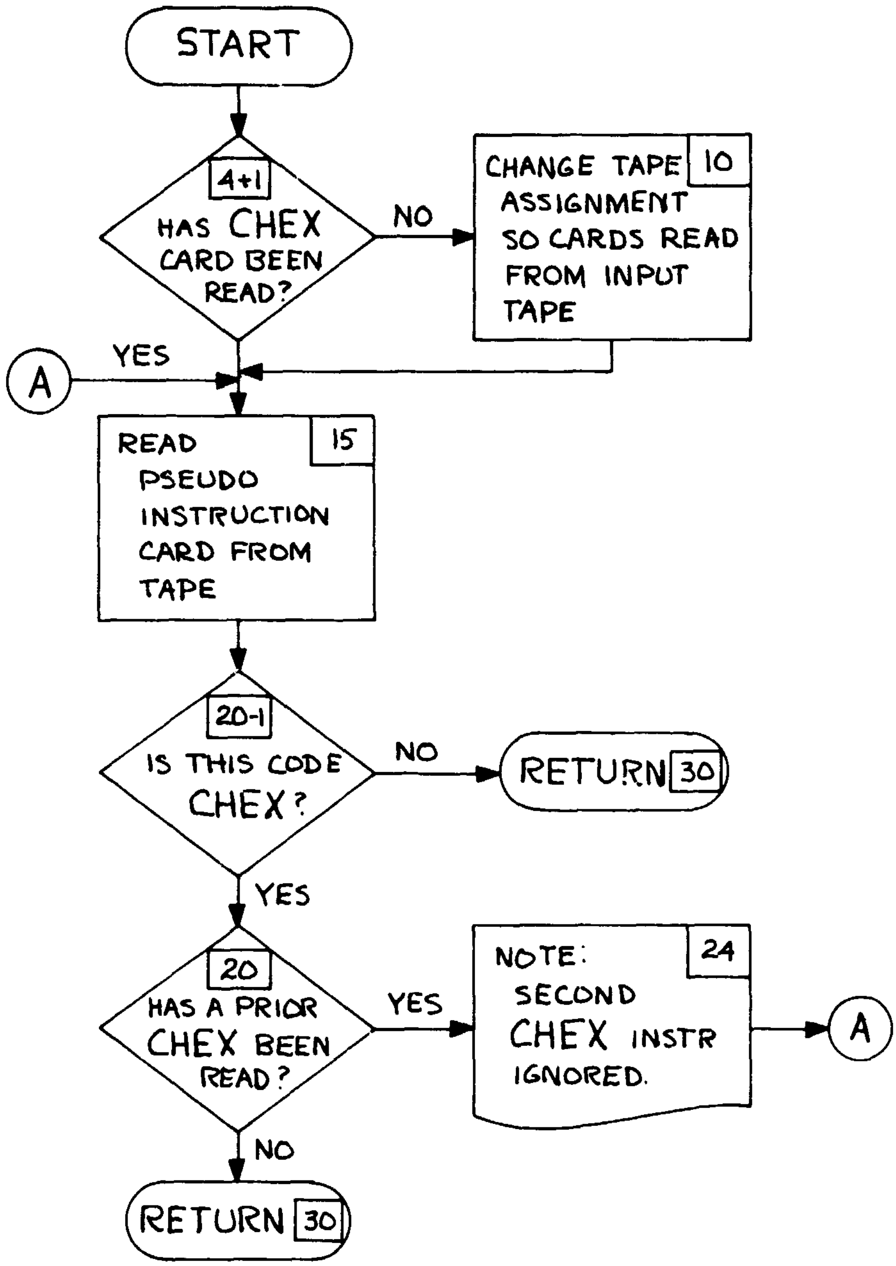



Figure 8-3.3 "RE" FLOW

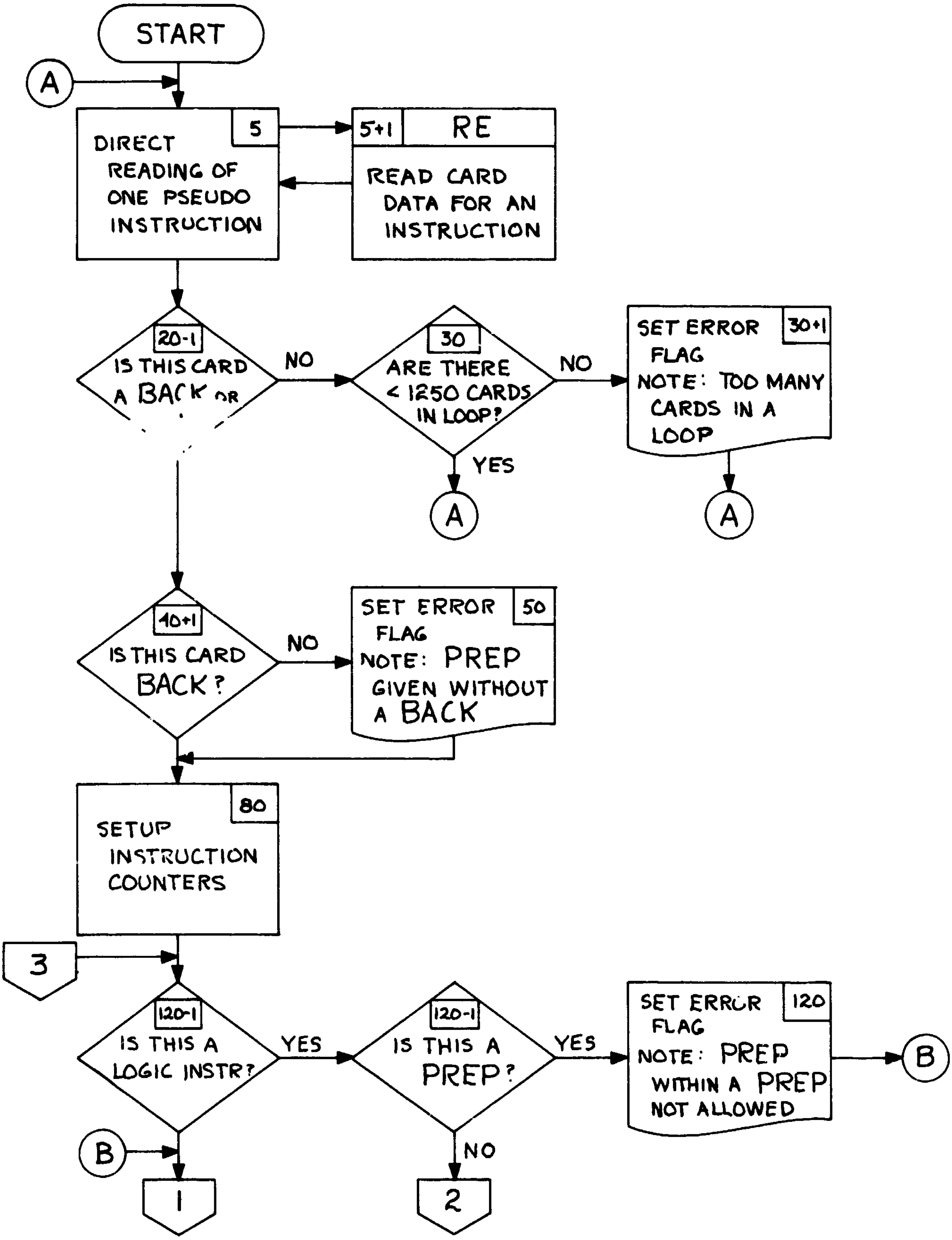



Figure 8-3.4 "PR" FLOW

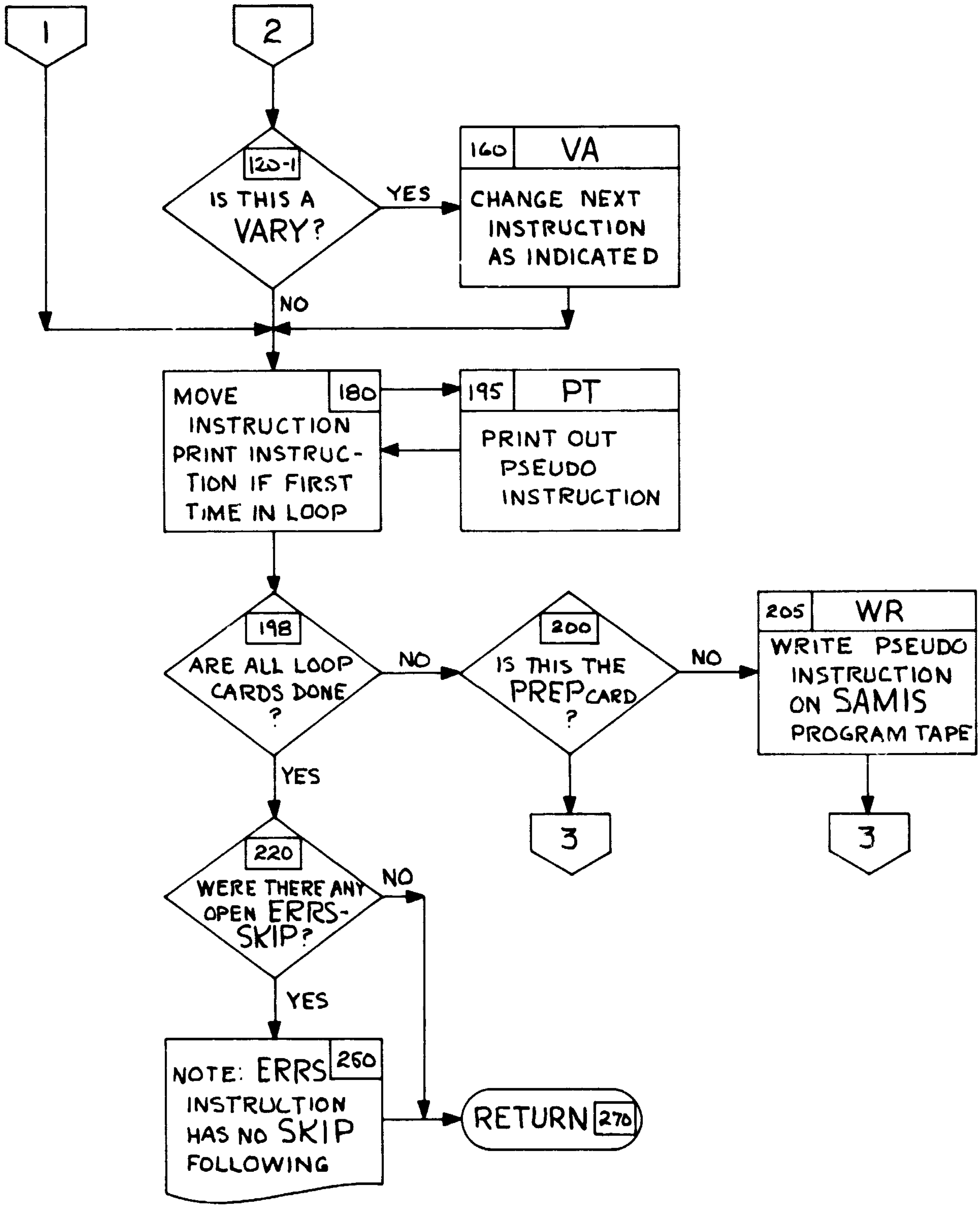



Figure 8-3.4 "PR" FLOW

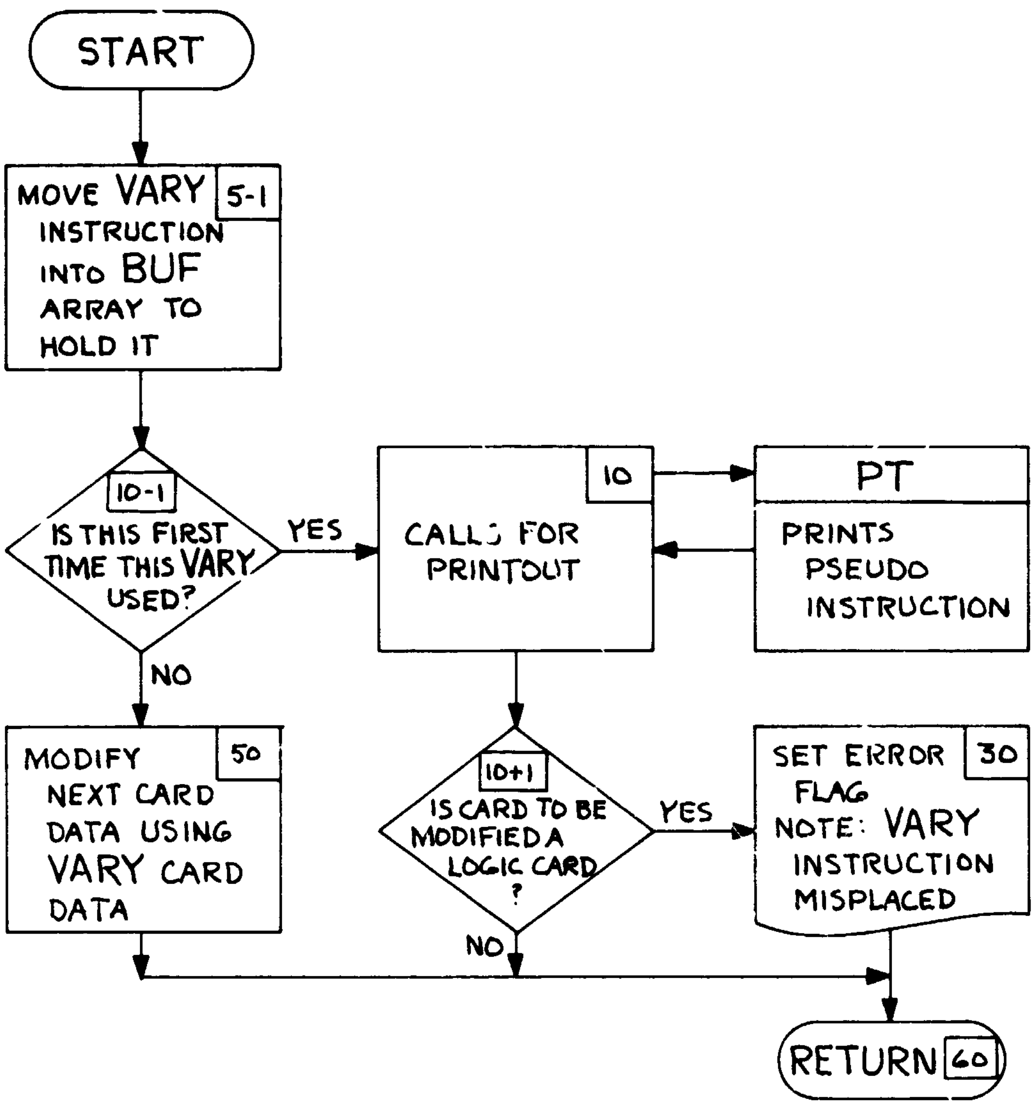



Figure 8-3.5 "VA" FLOW

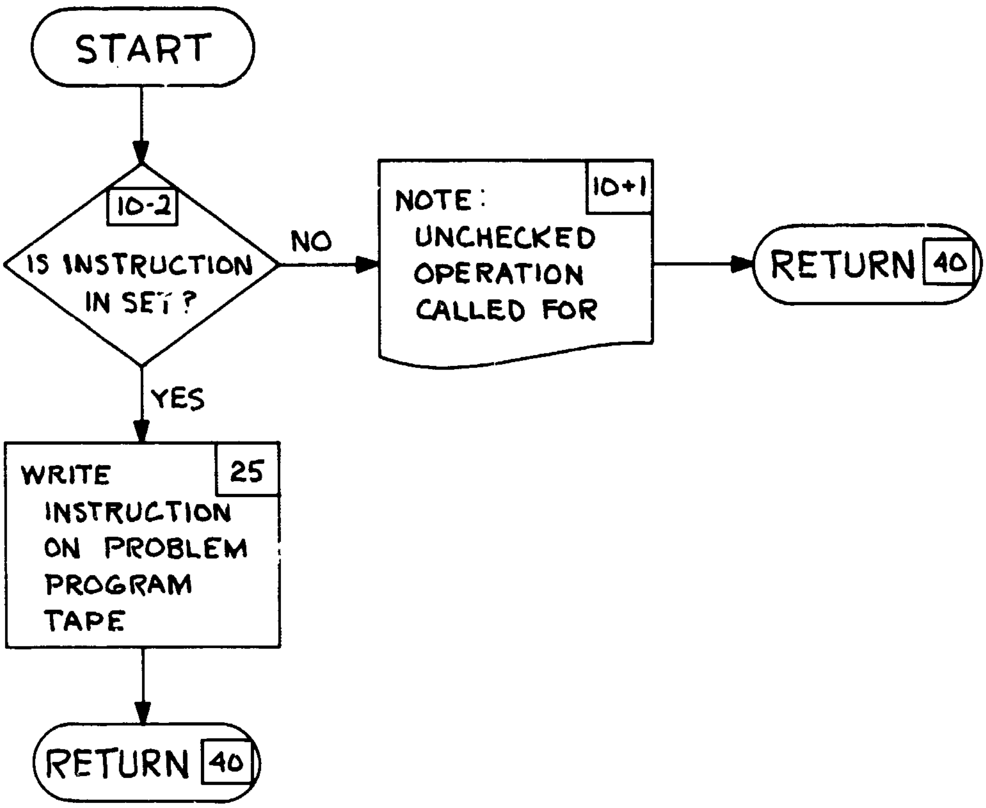



Figure 8-3.6 "WR" FLOW

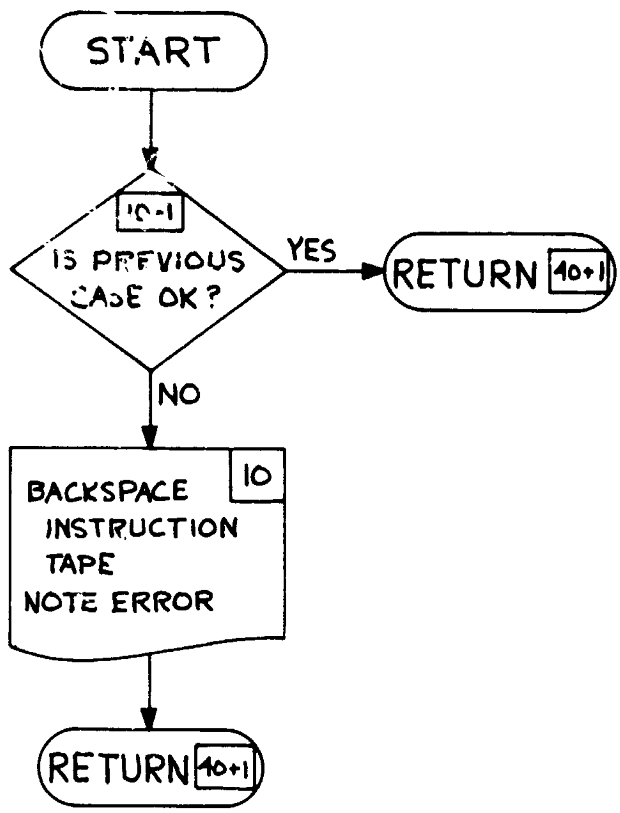



Figure 8-3.7 "BS" FLOW

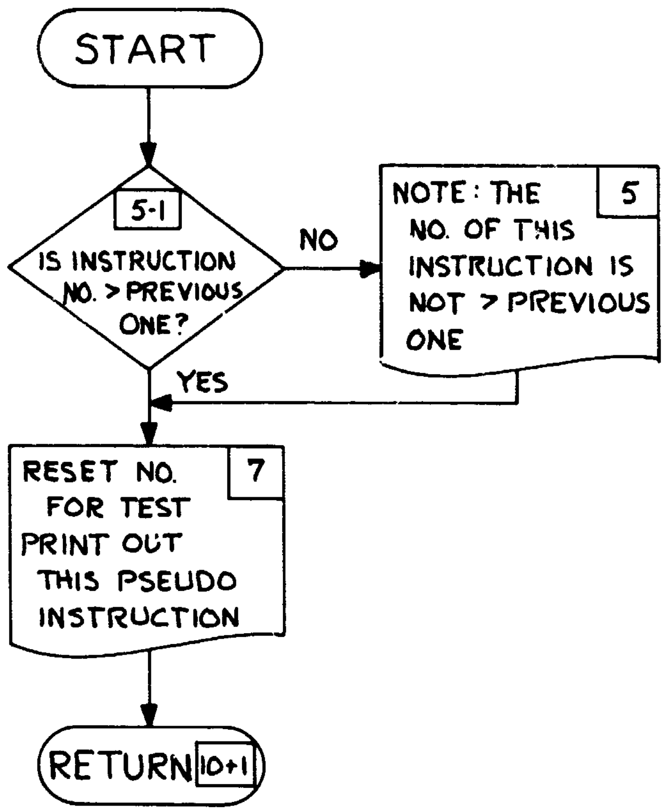



Figure 8-3.8 "PT" FLOW

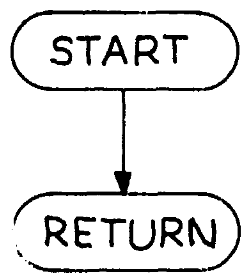



Figure 8-3.9 "MINTCK" FLOW

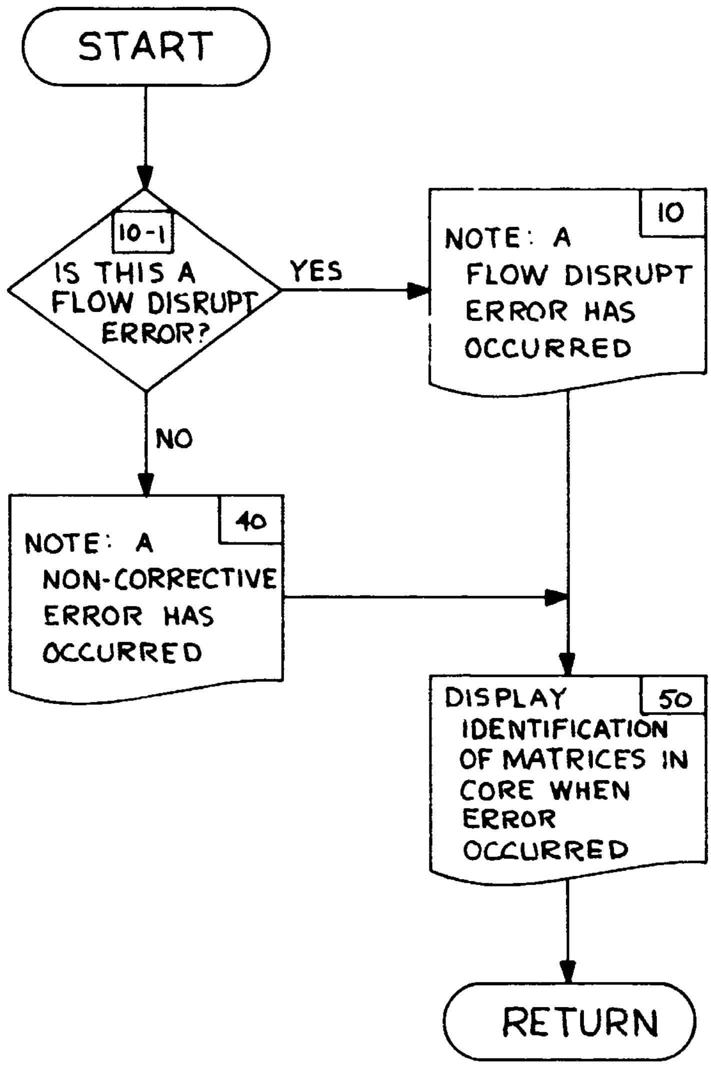



Figure 8-4.1 "ERROR" FLOW

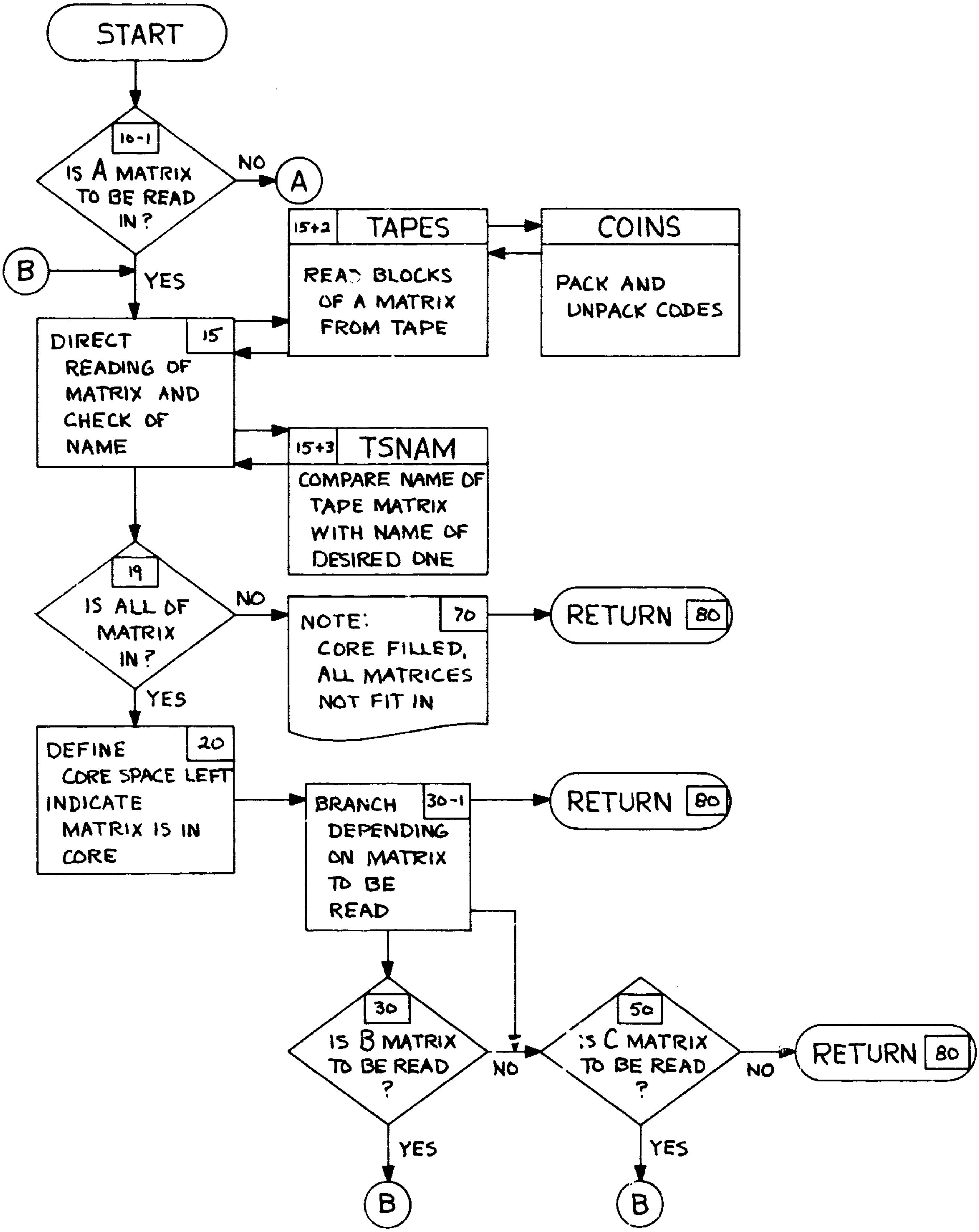



Figure 8-4.2 "IN" FLOW

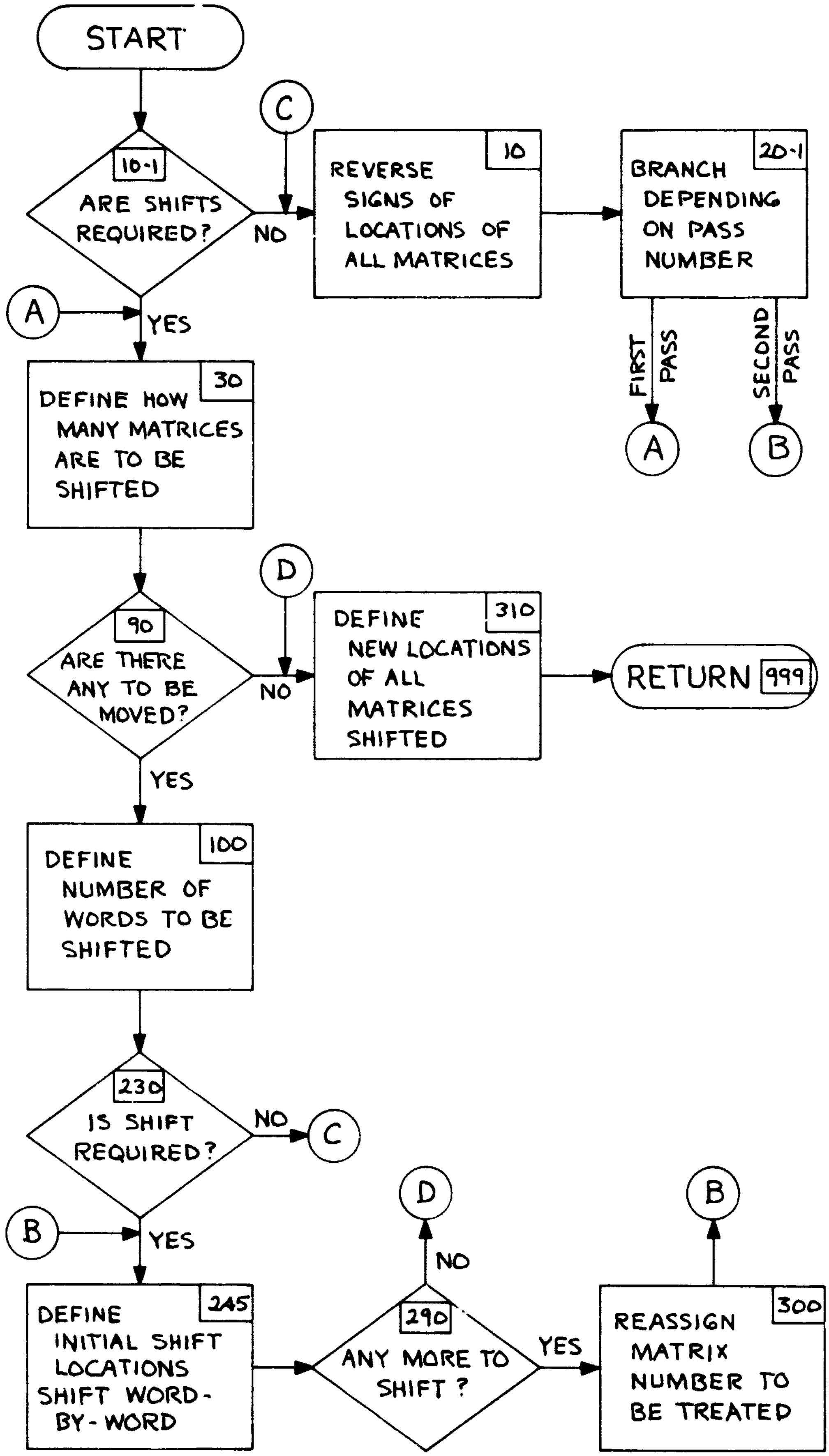



Figure 8-4.3 "SHIFT" FLOW

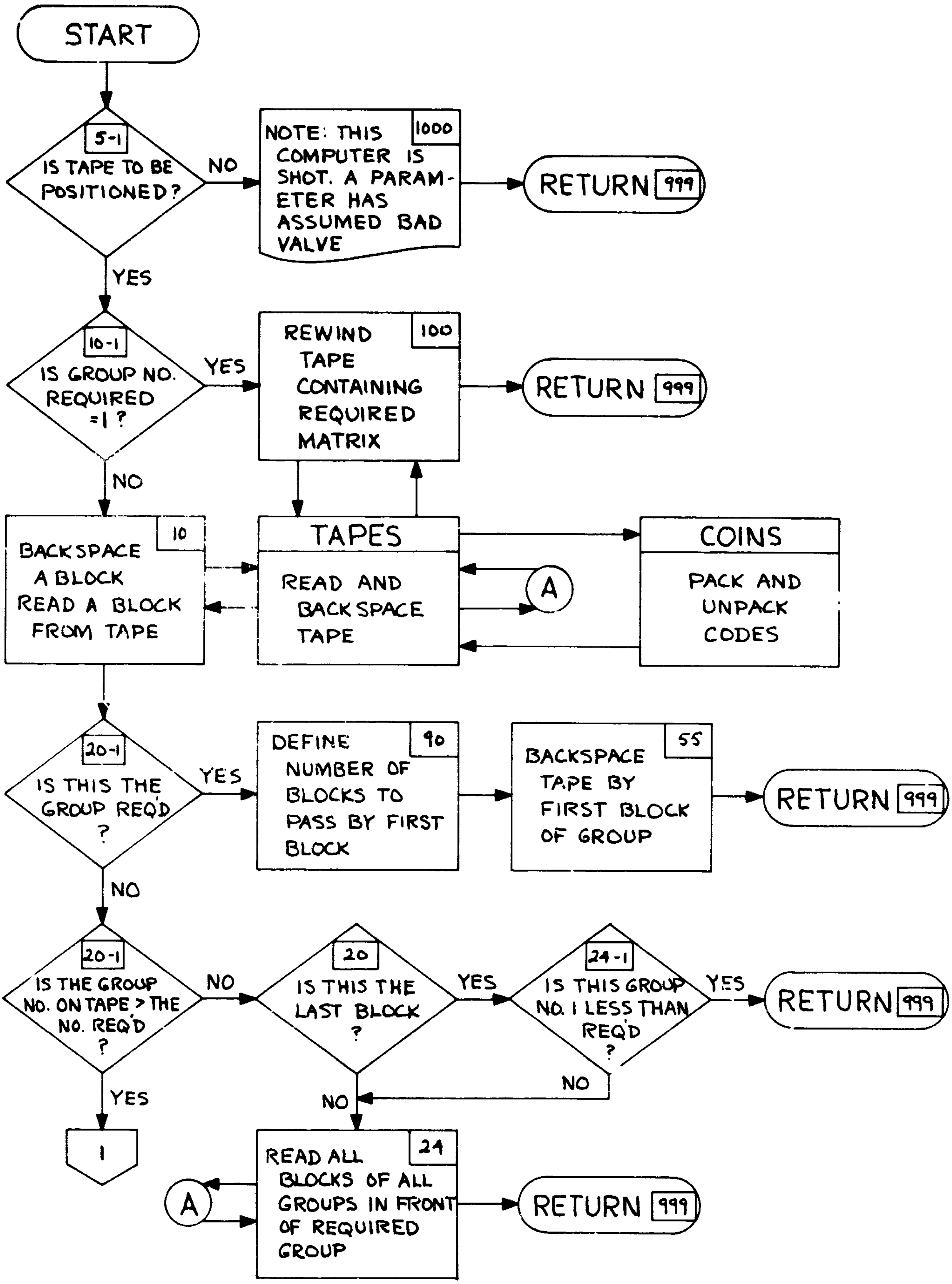



Figure 8-4.4 FINDS FLOW

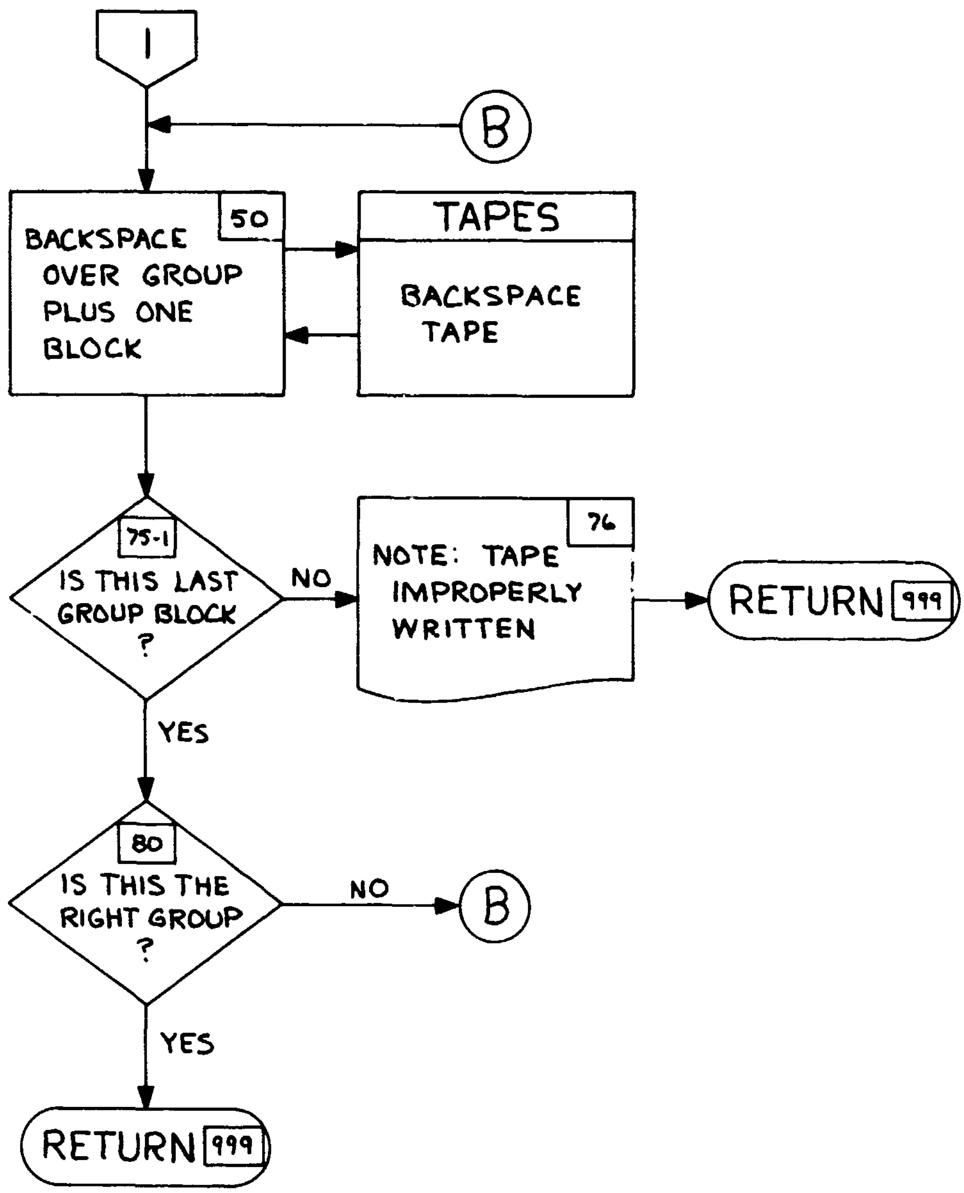



Figure 8-4.4 "FINDS" FLOW

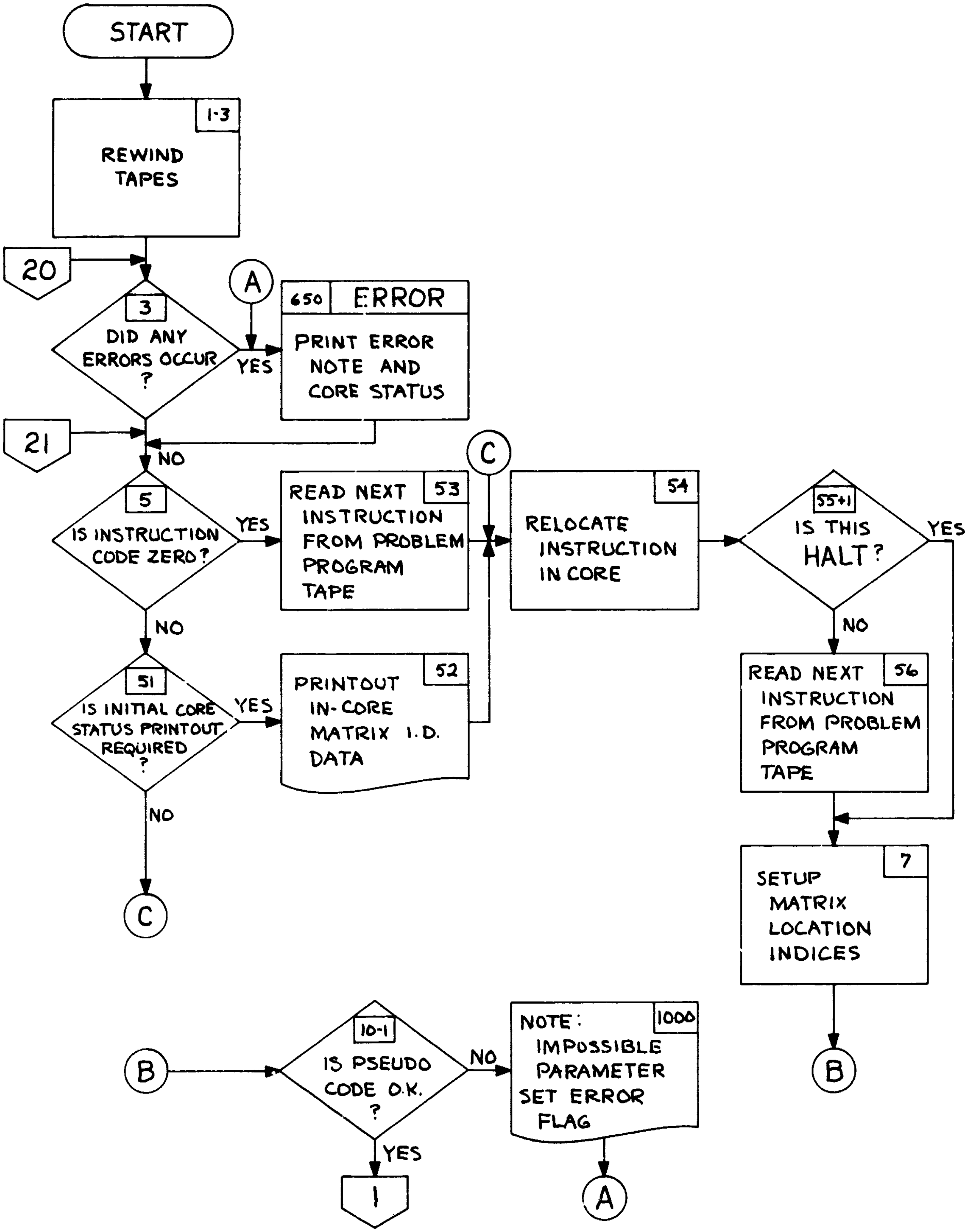



Figure 8-4.5 "MINTS" FLOW

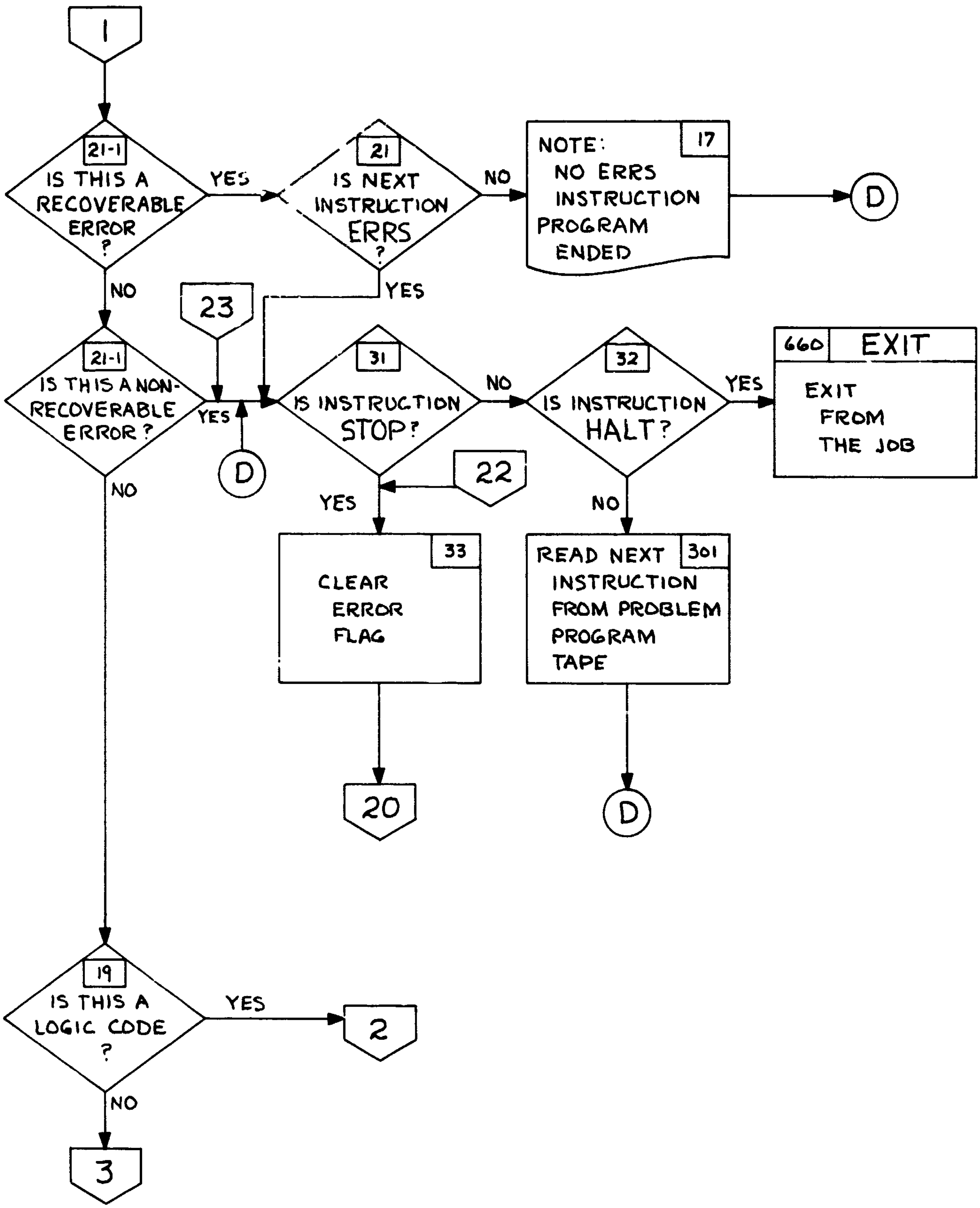



Figure 8-4.5 "MINTS" FLOW

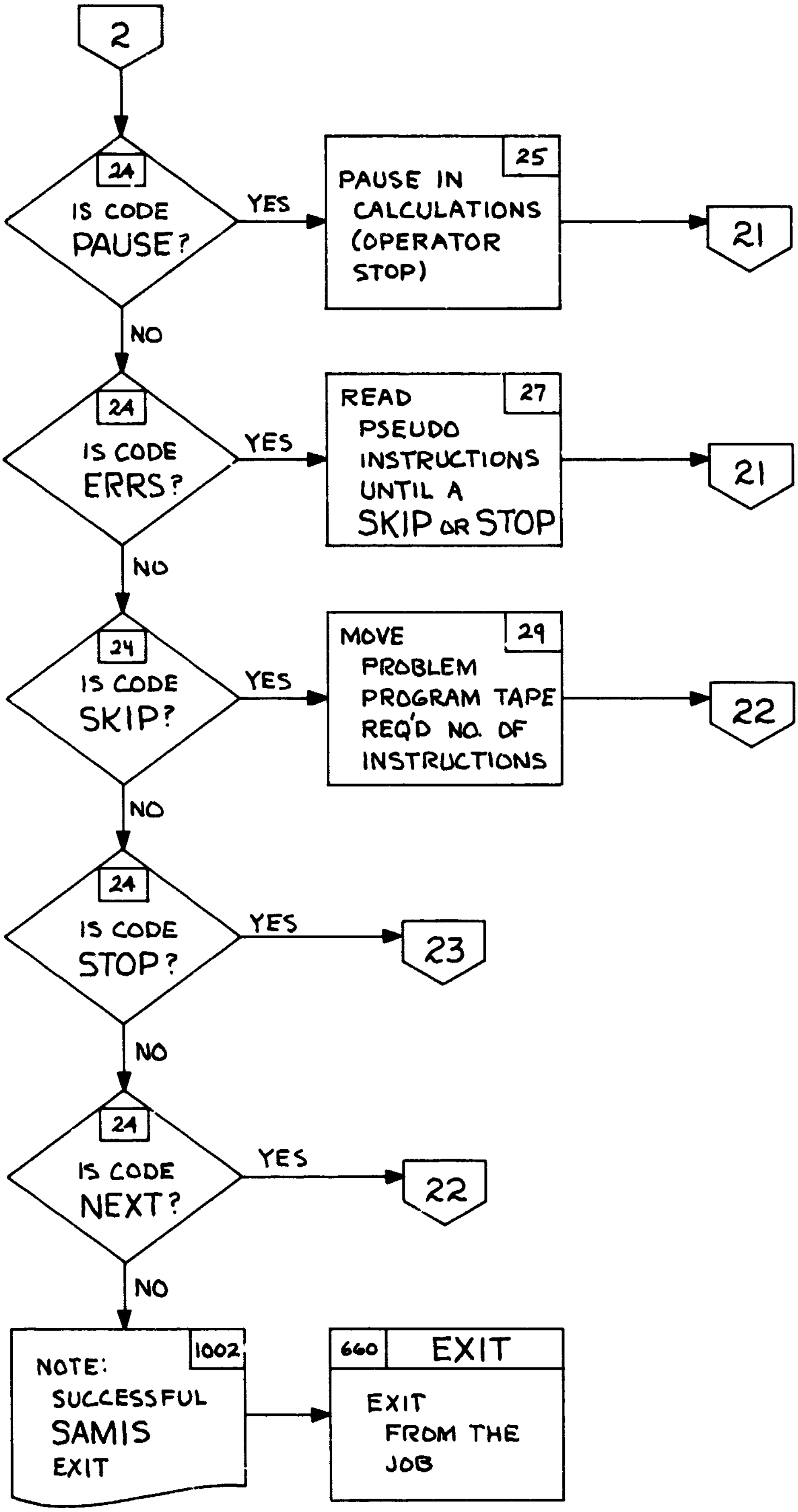



Figure 8-4. "MINTS" FLOW

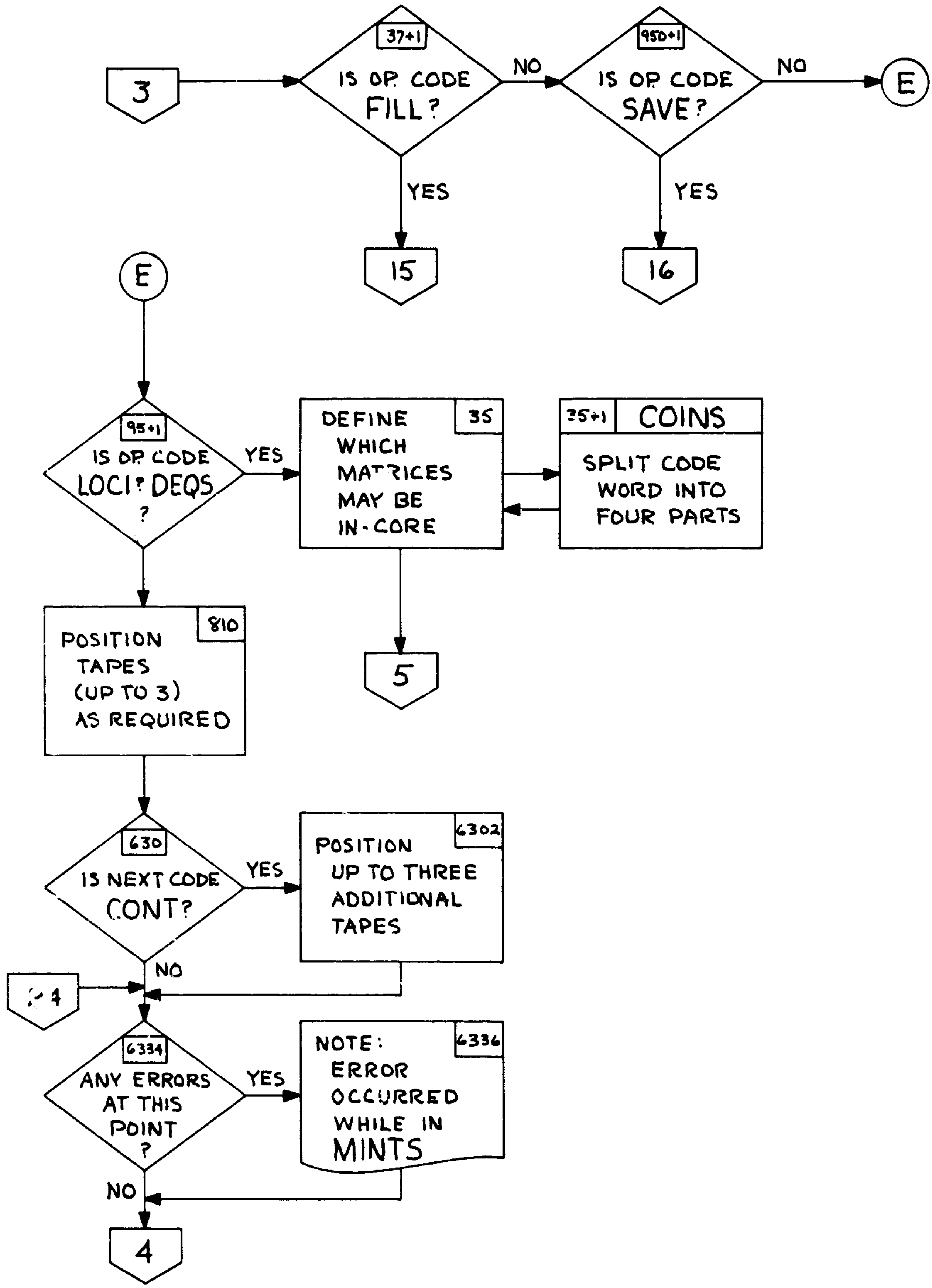



Figure 8-4.5 "MINTS" FLOW

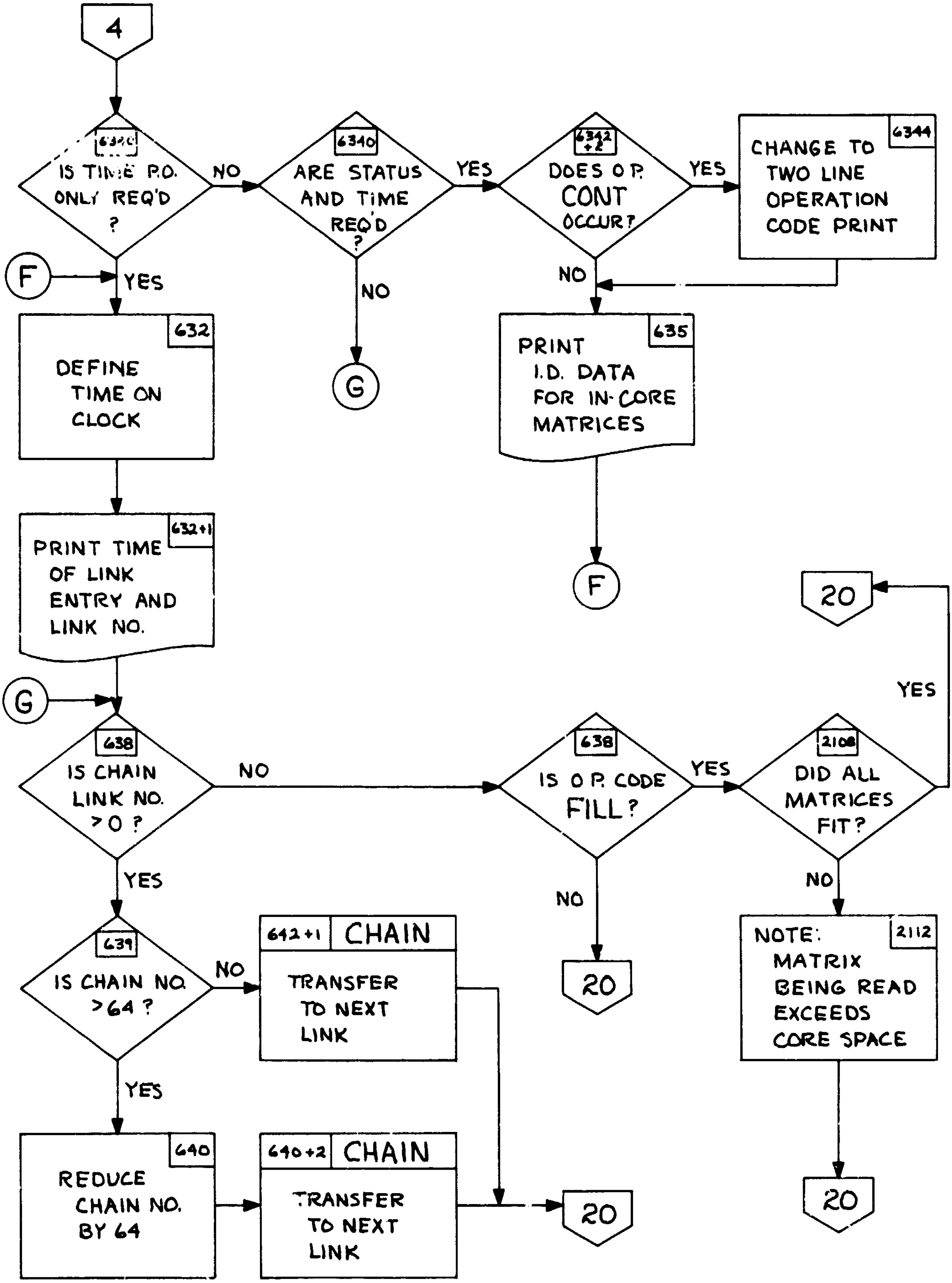



Figure 8-4.5 "MINTS" FLOW

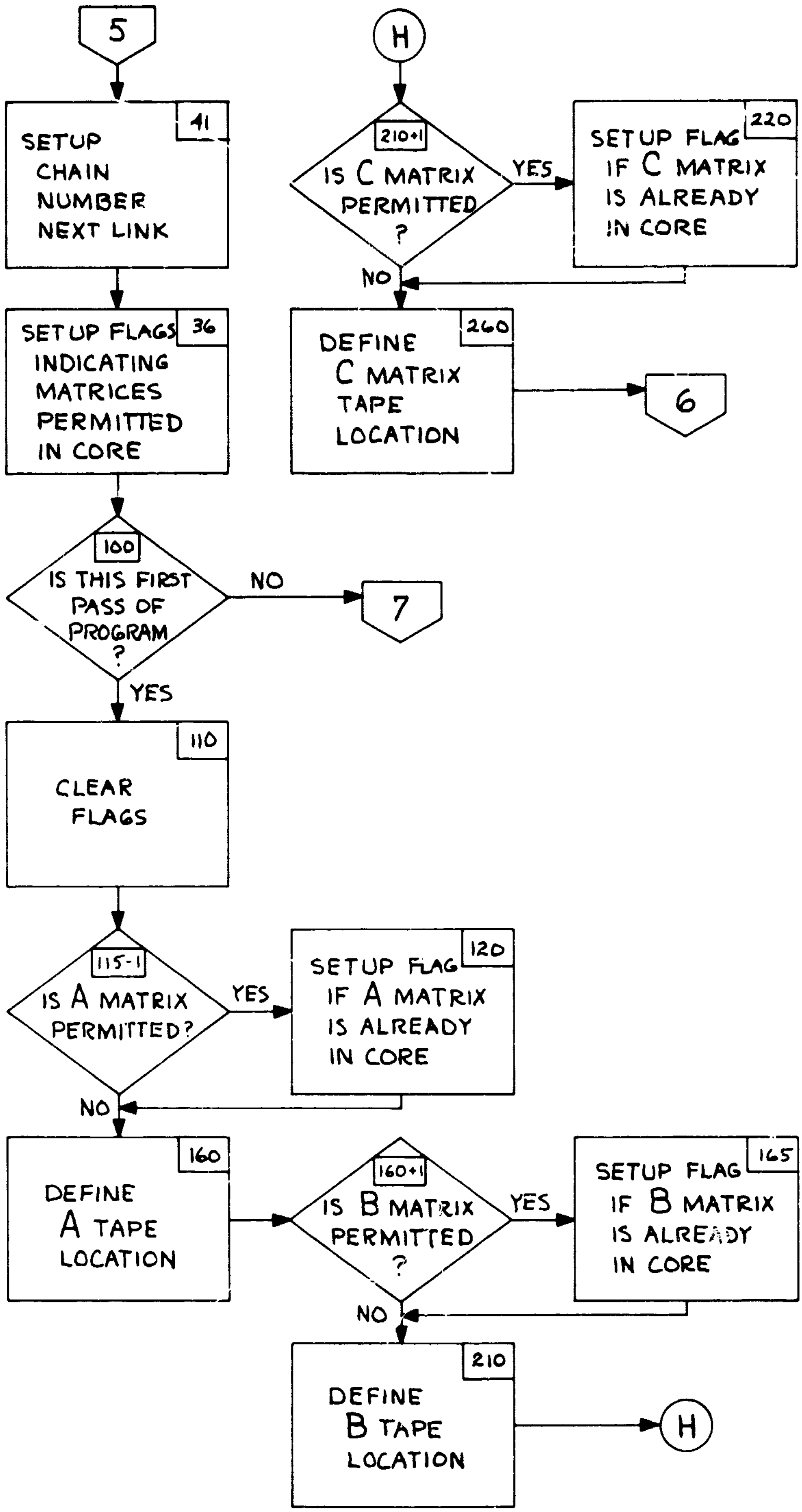



Figure 8-4.5 "MINTS" FLOW

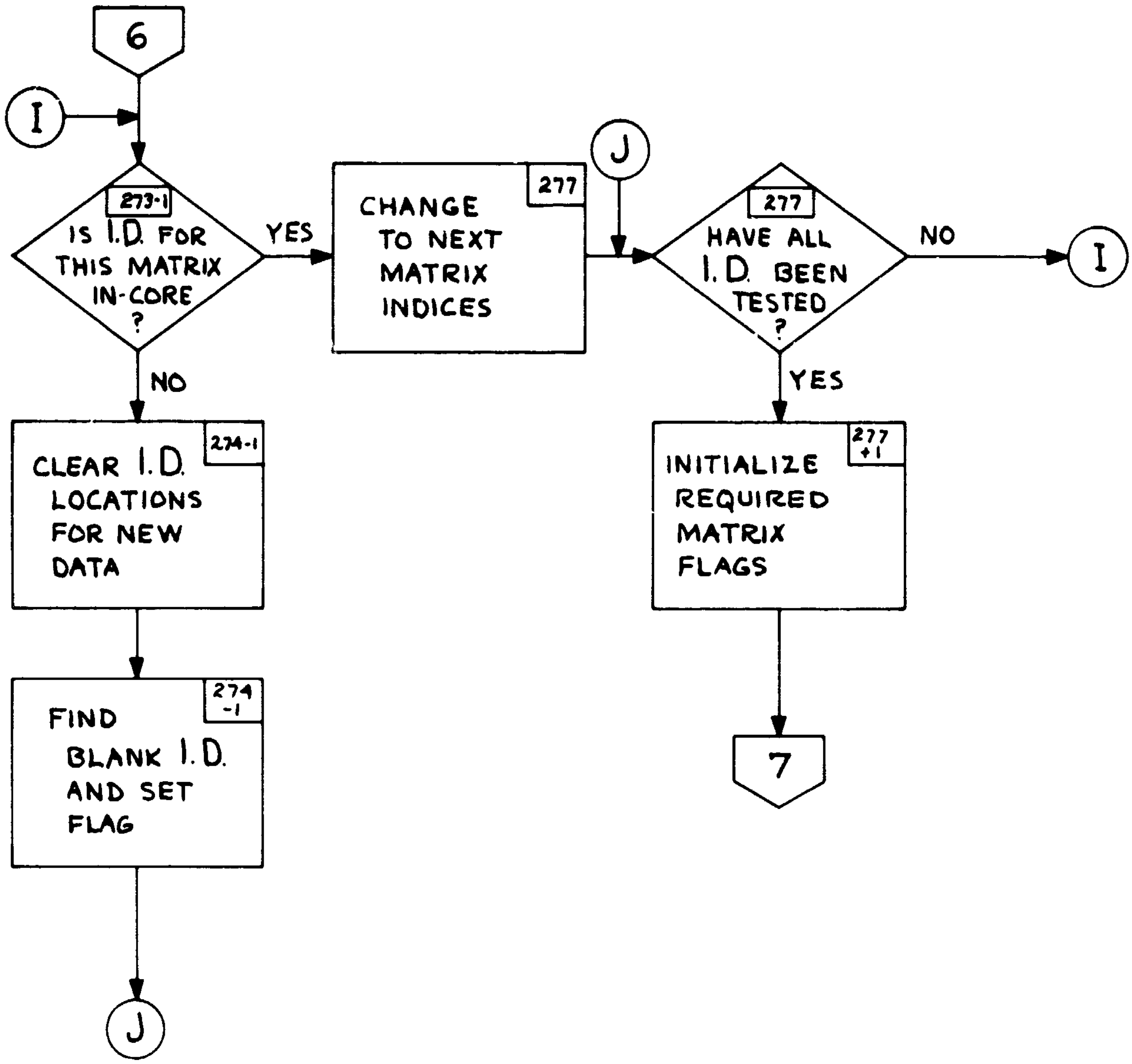



Figure 8-4.5 "MINTS" FLOW

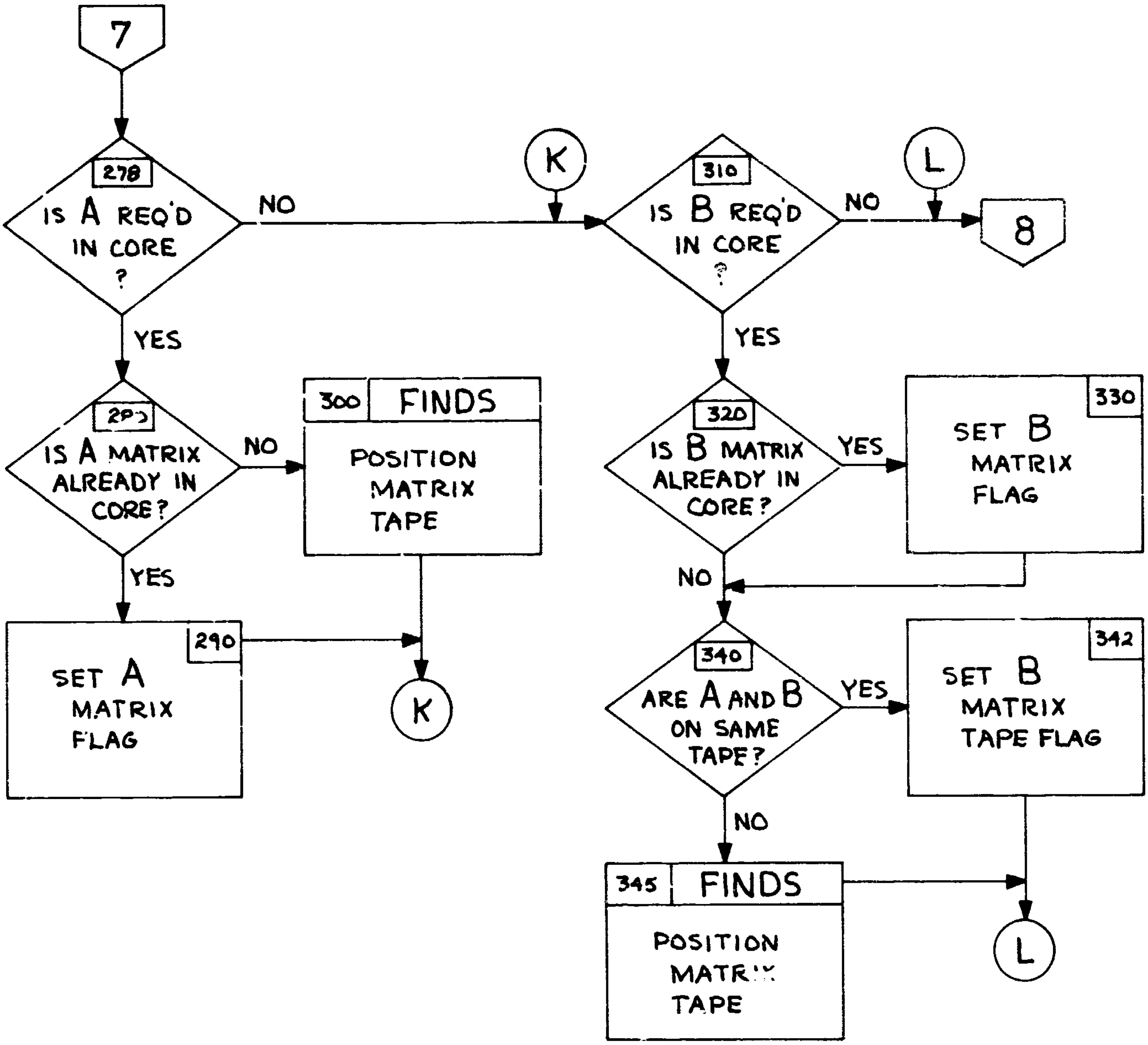



Figure 8-4.5 "MINTS" FLOW

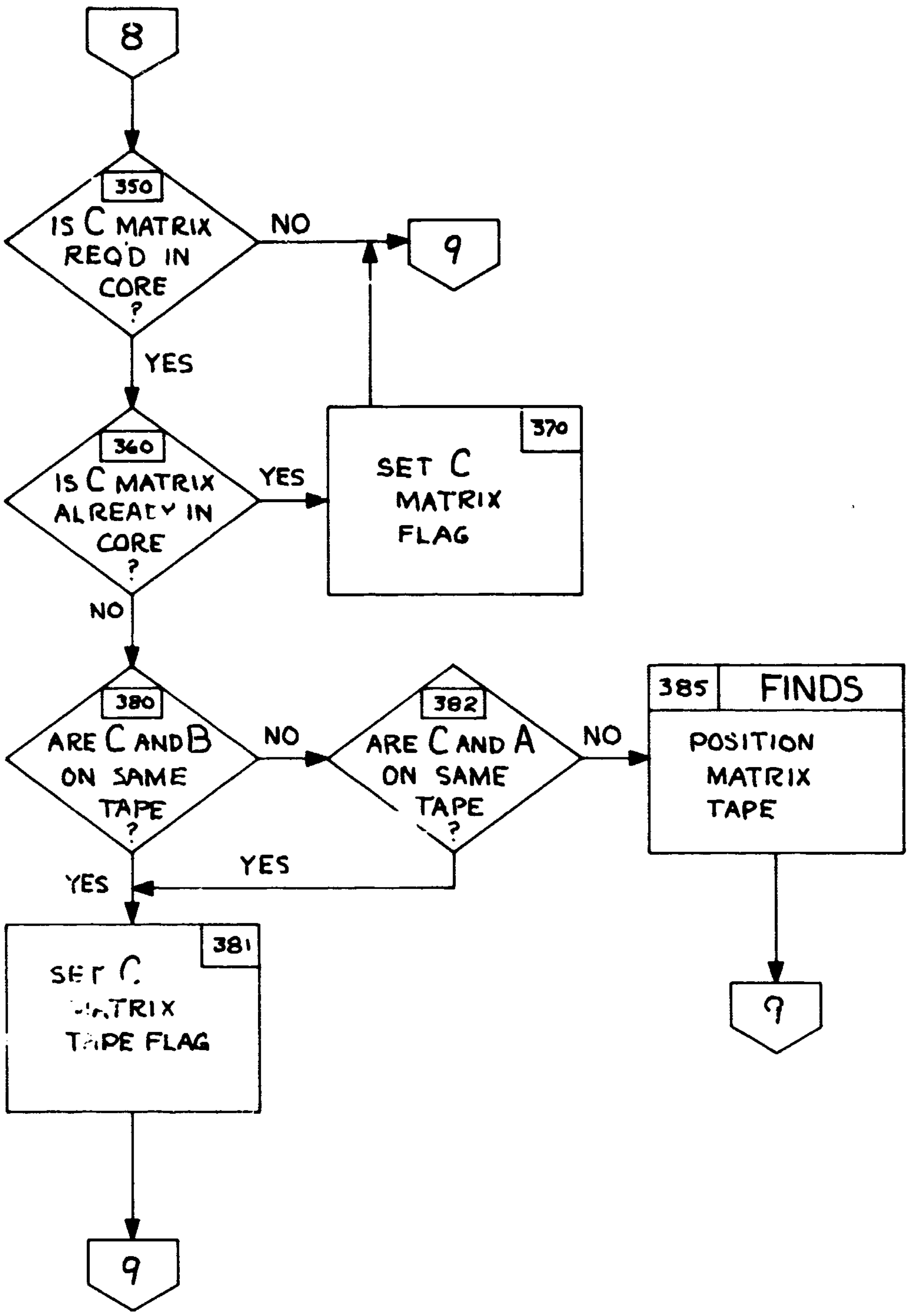



Figure 8-4.5 "MINTS" FLOW

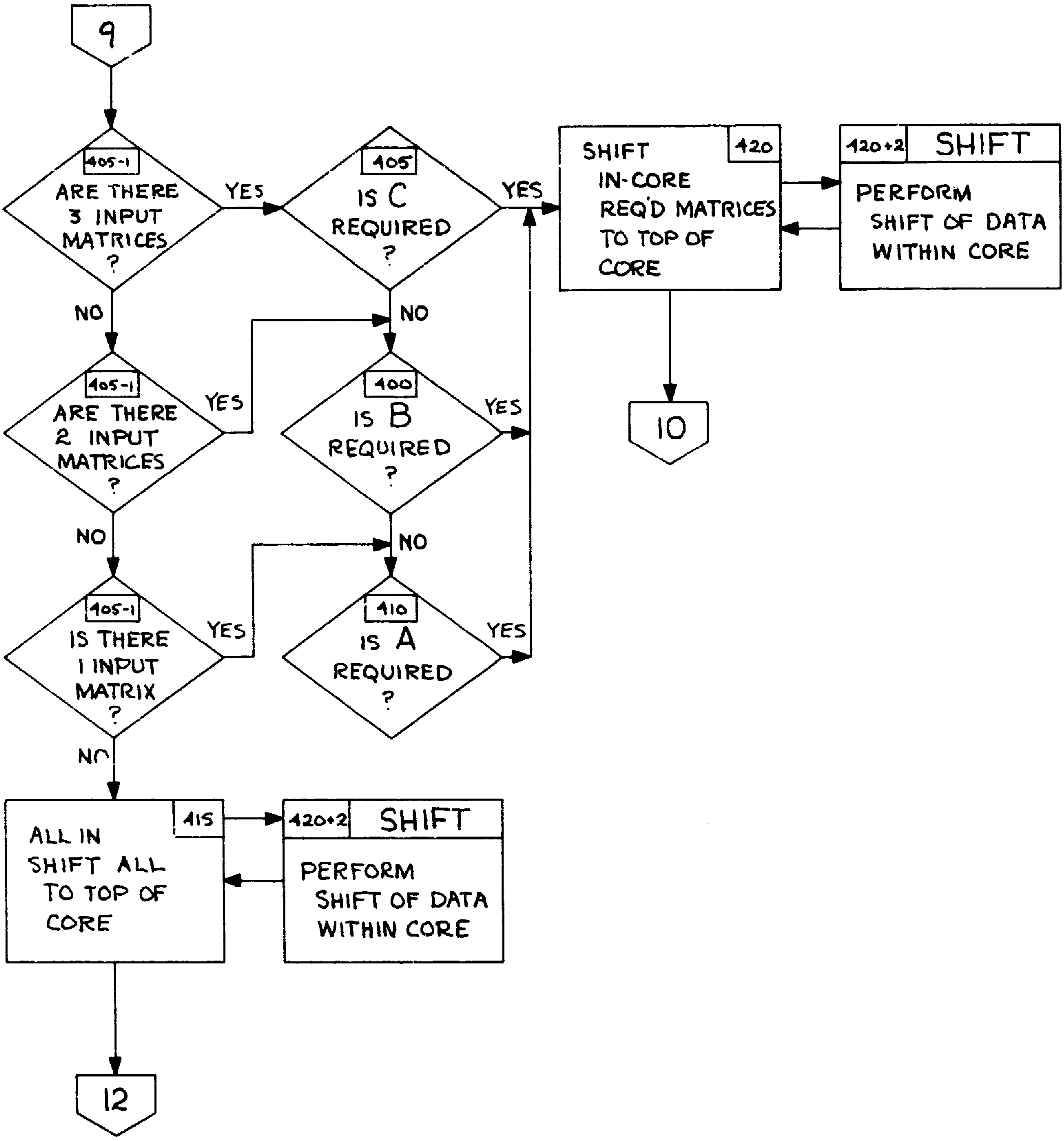



Figure 8-4.5 "MINTS" FLOW

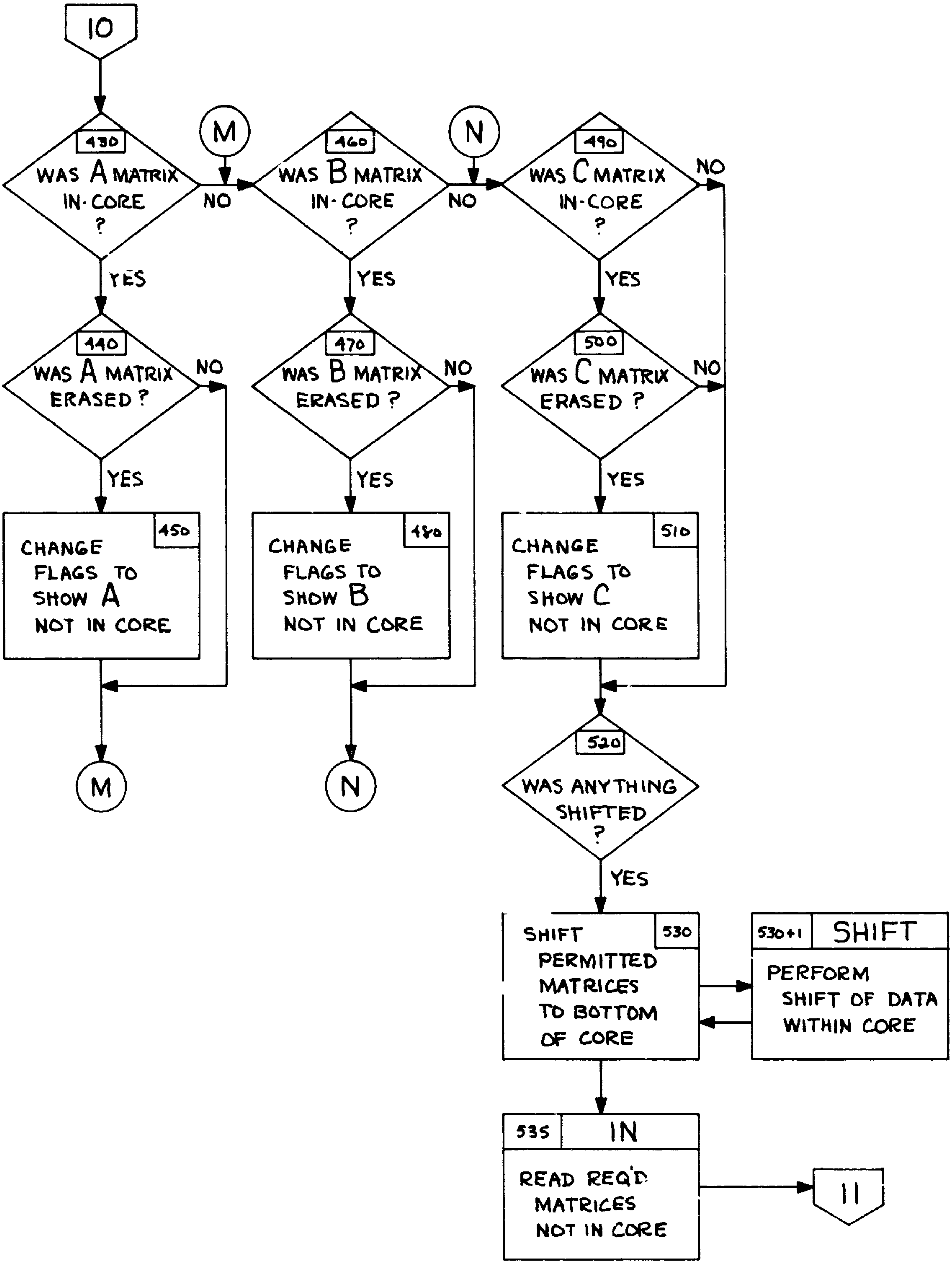



Figure 8-4.5 "MINTS" FLOW

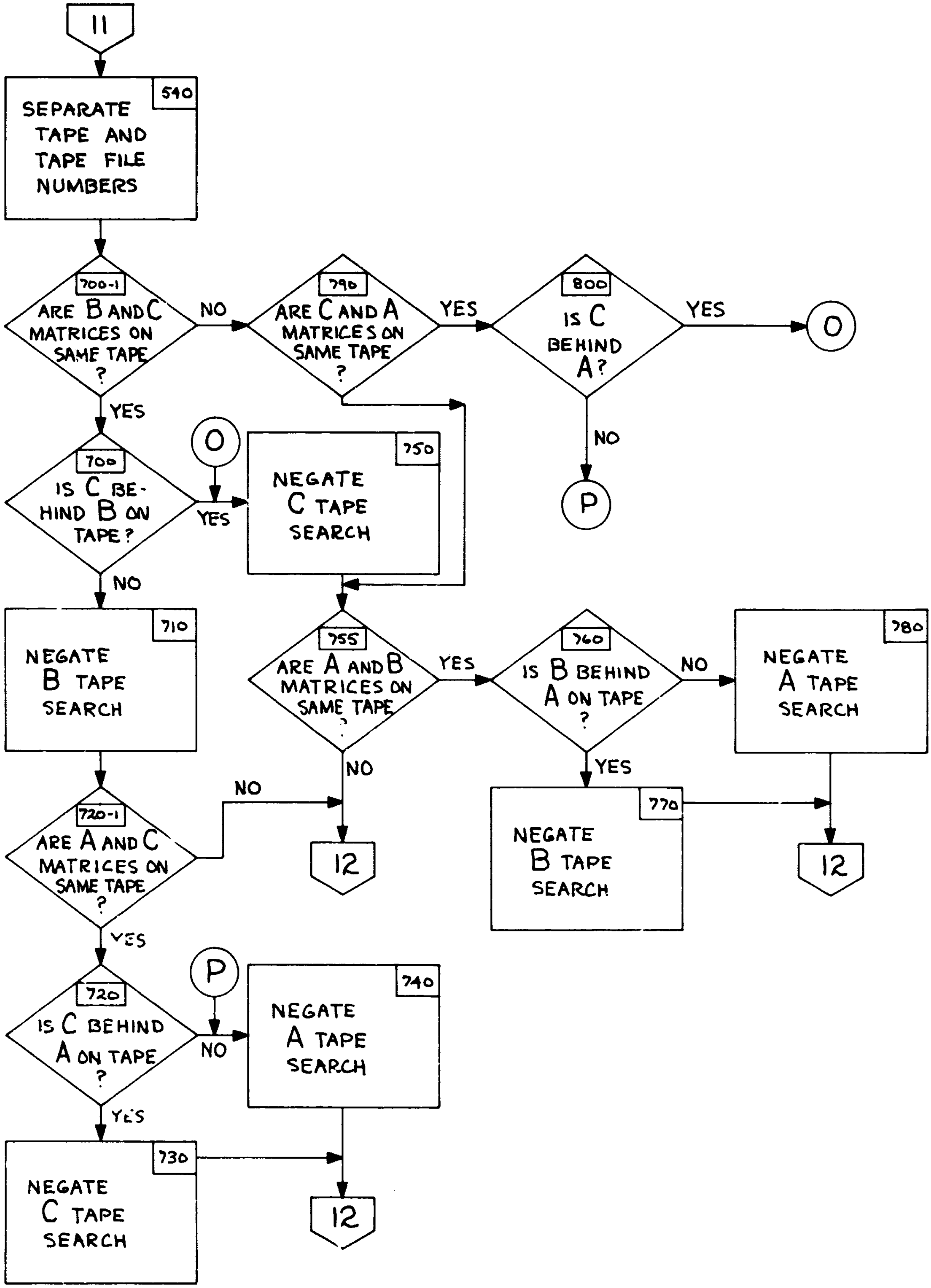



Figure 8-4.5 "MINTS" FLOW

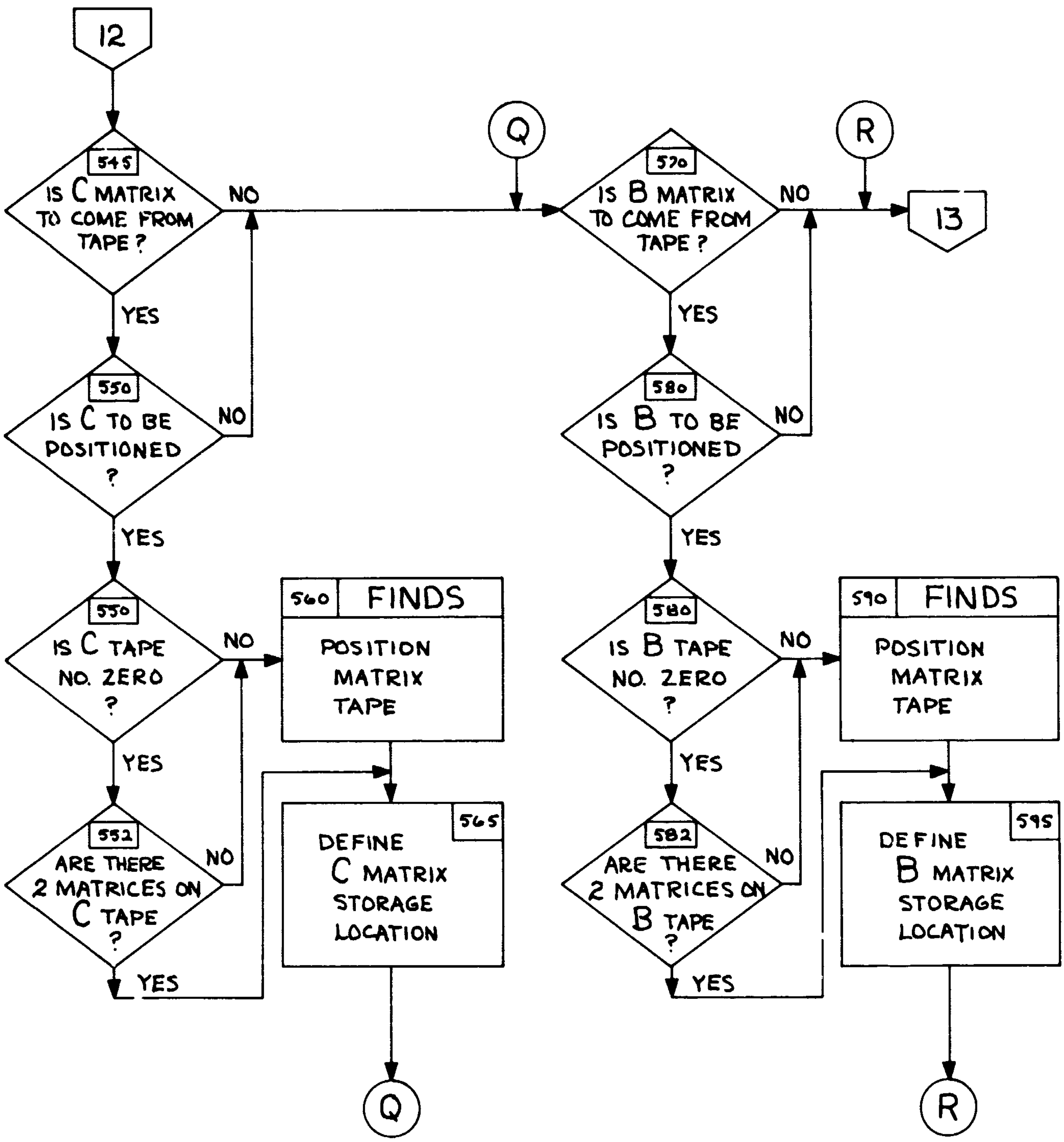



Figure 8-4.5 "MINTS" FLOW

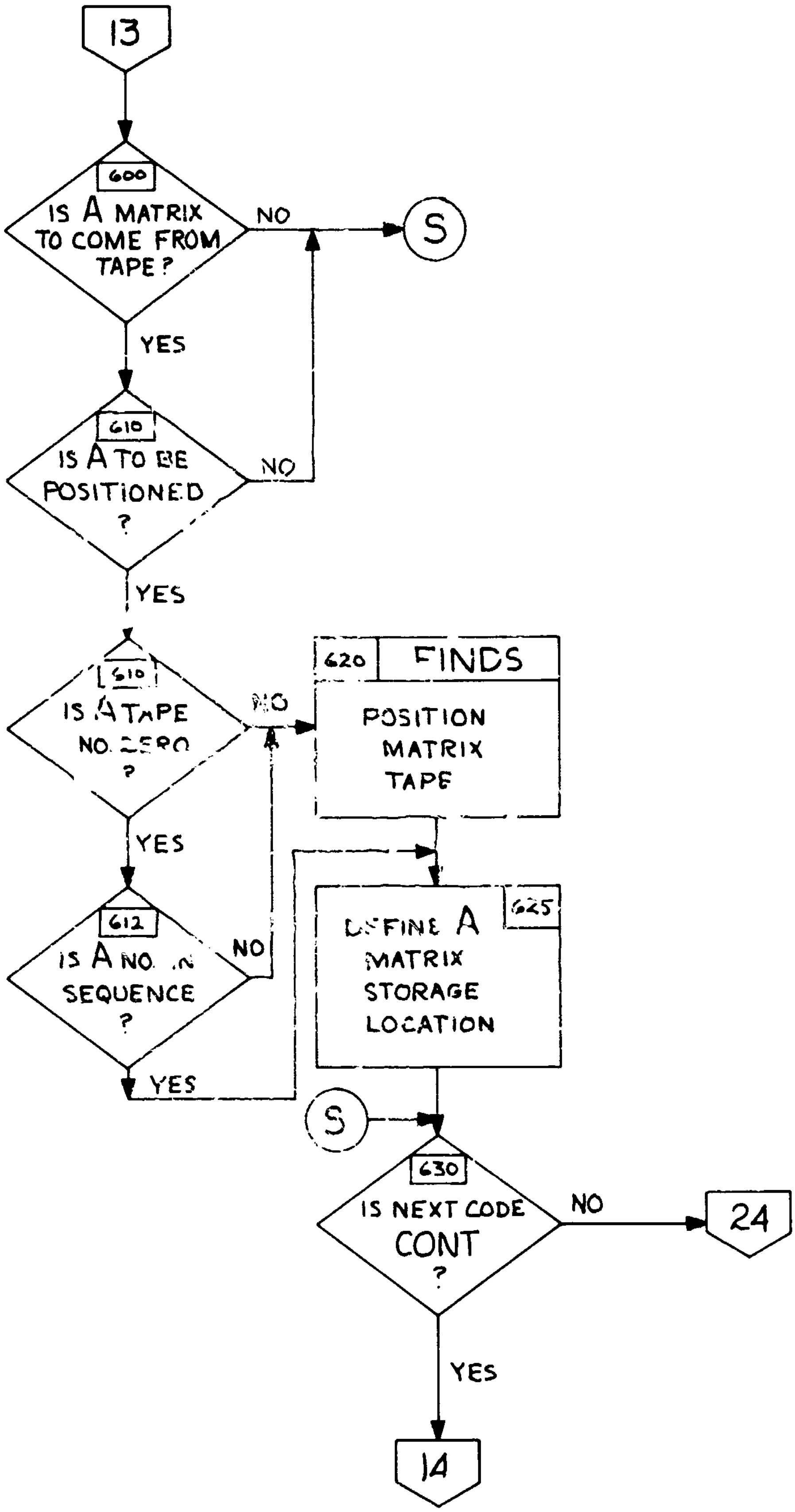



Figure 8-4.5 "MINTS" FLOW

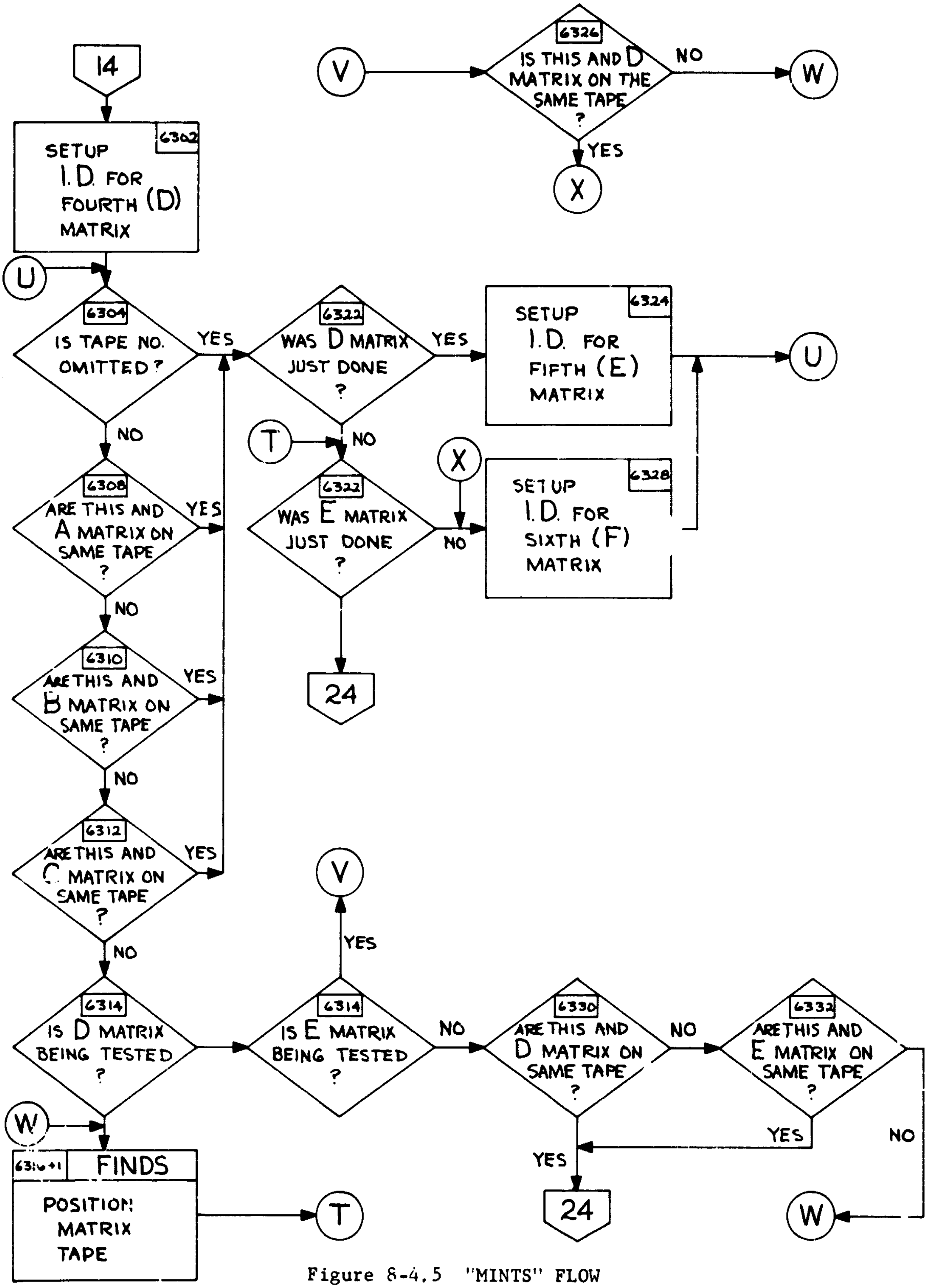



Figure 8-4.5 "MINTS" FLOW

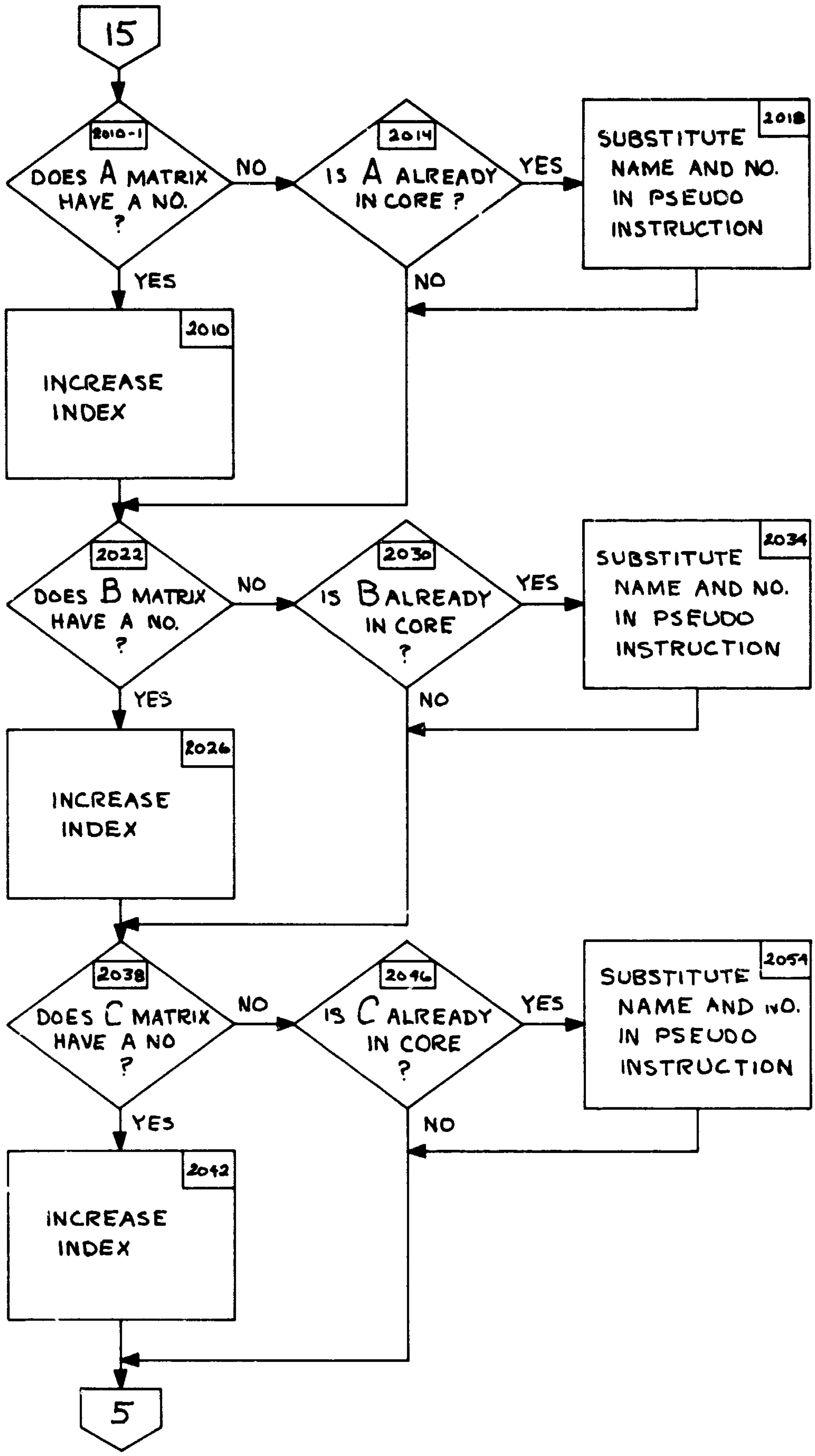



Figure 8-4.5 "MINTS" FLOW

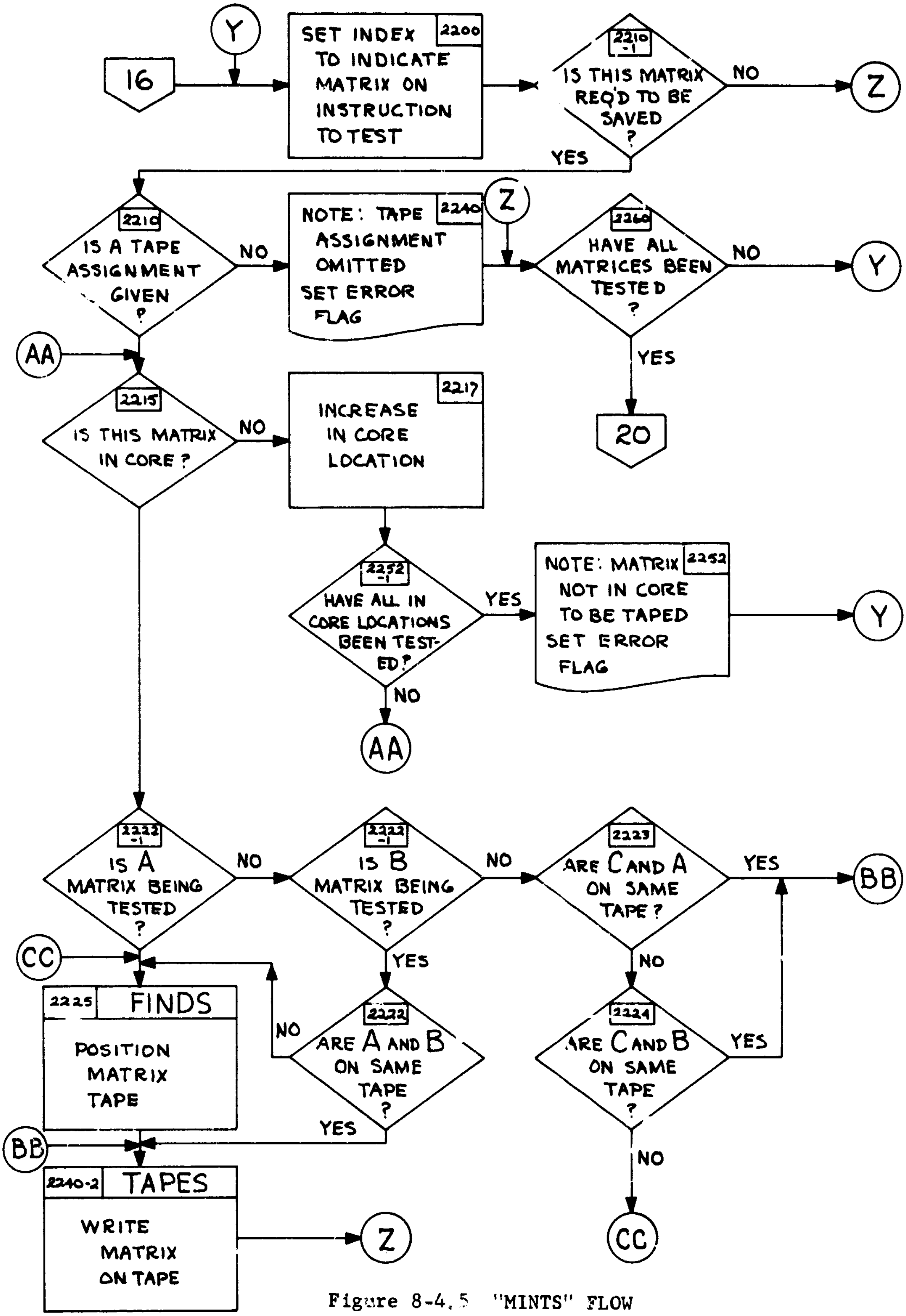



Figure 8-4.5 "MINTS" FLOW

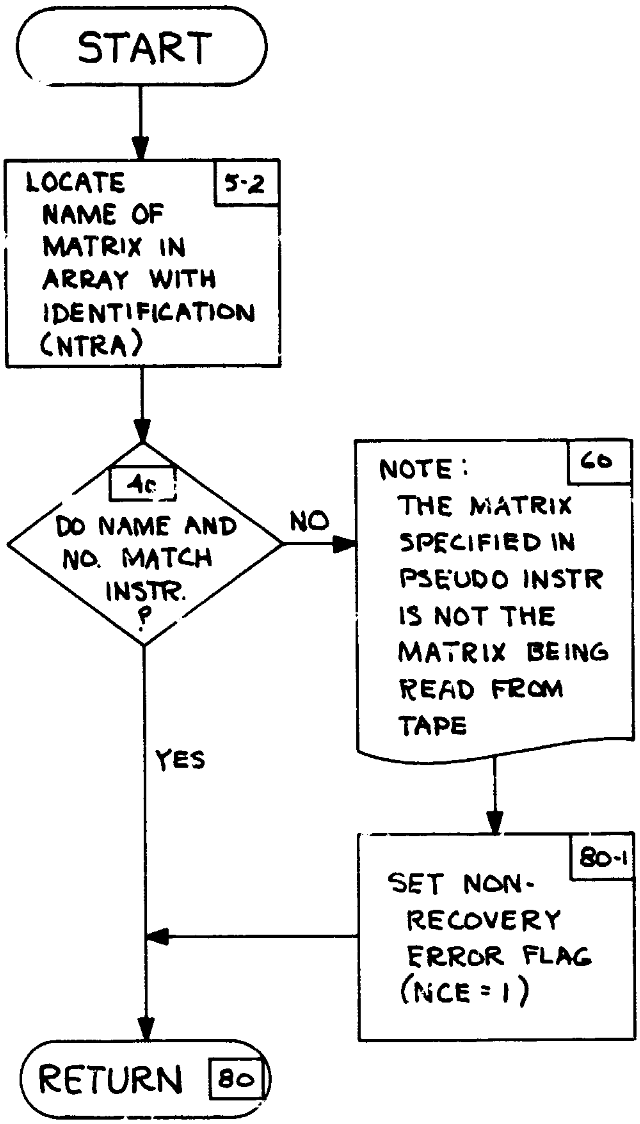



Figure 8-4.6 "TSNAM" FLOW

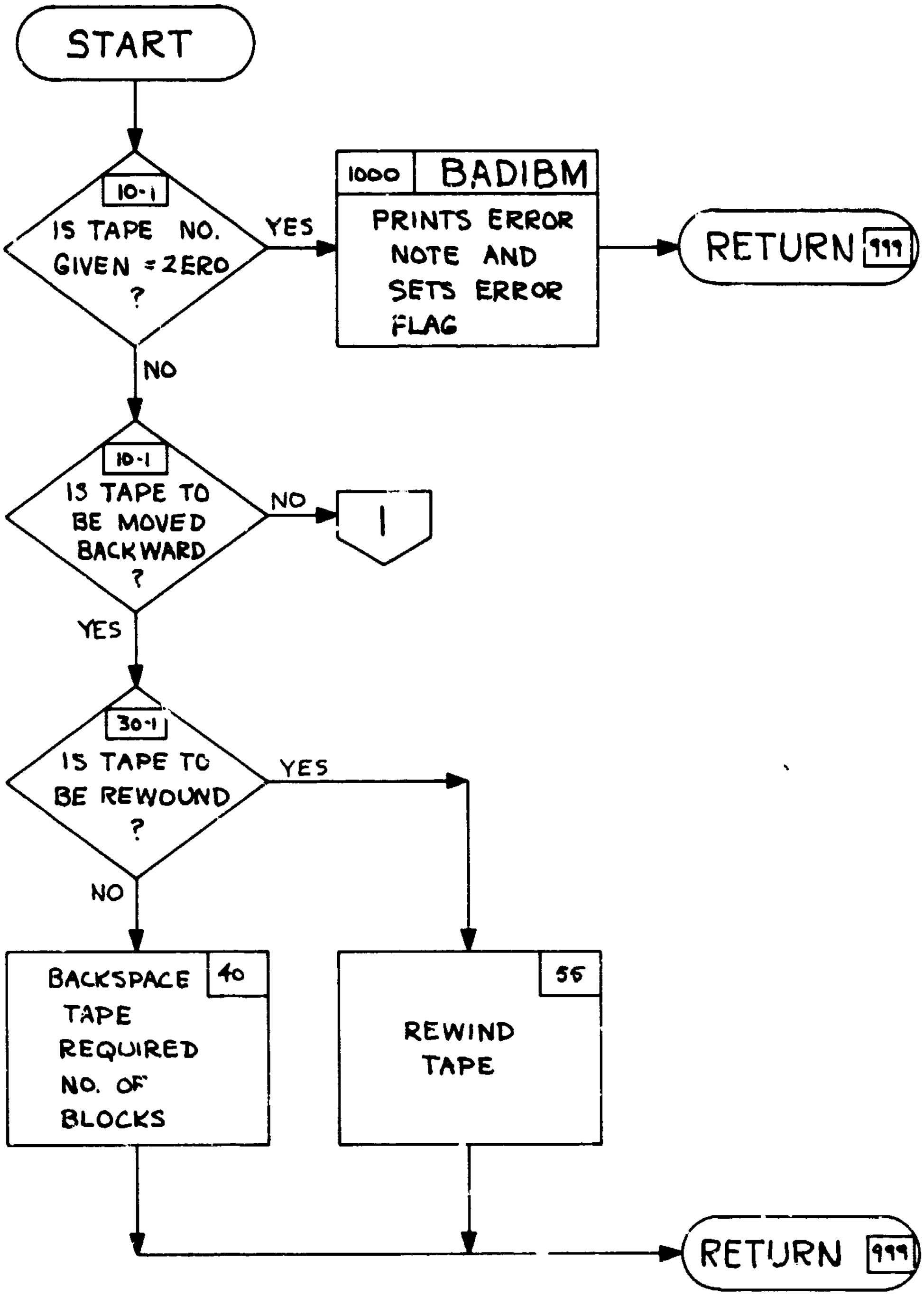



Figure 8-4.7 "TAPES" FLOW

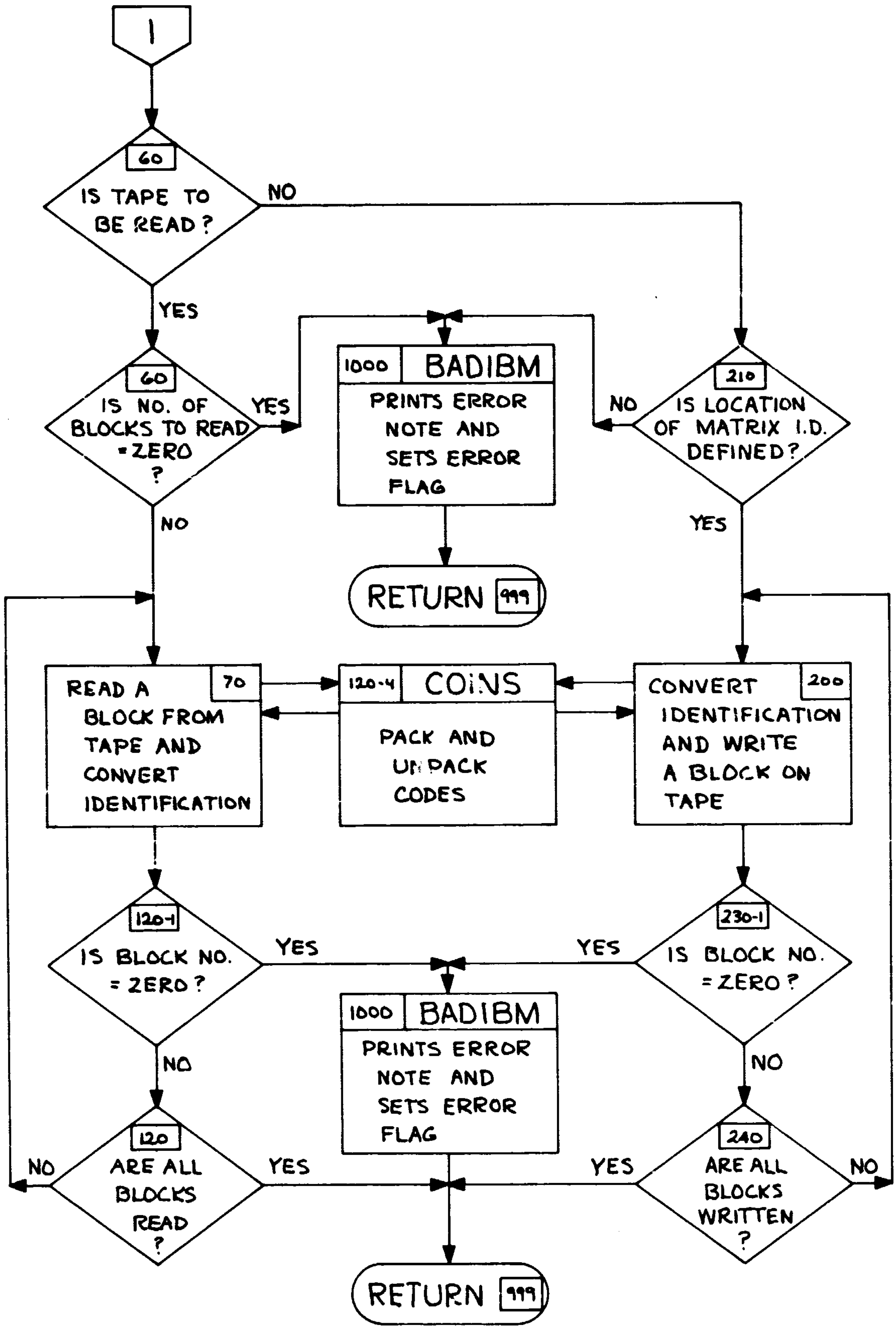



Figure 8-4.7 "TAPES" FLOW

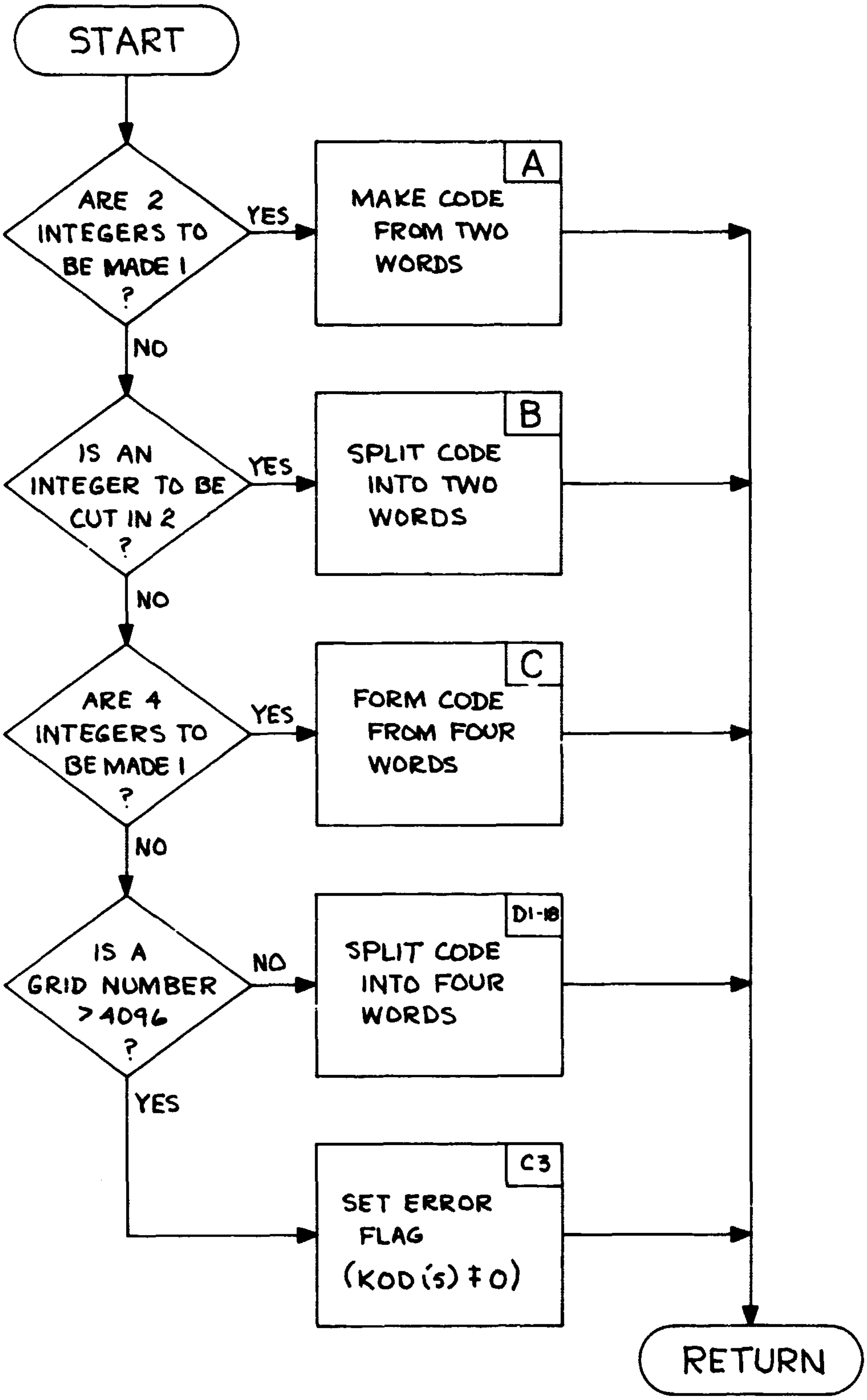



Figure 8-4.8 "COINS" FLOW

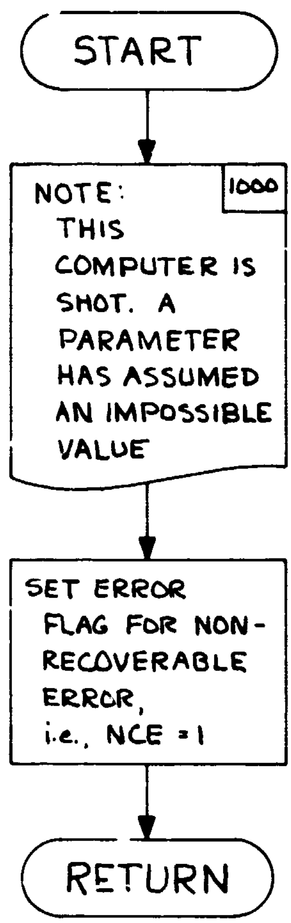



Figure 8-4.9 'BADIBM" FLOW

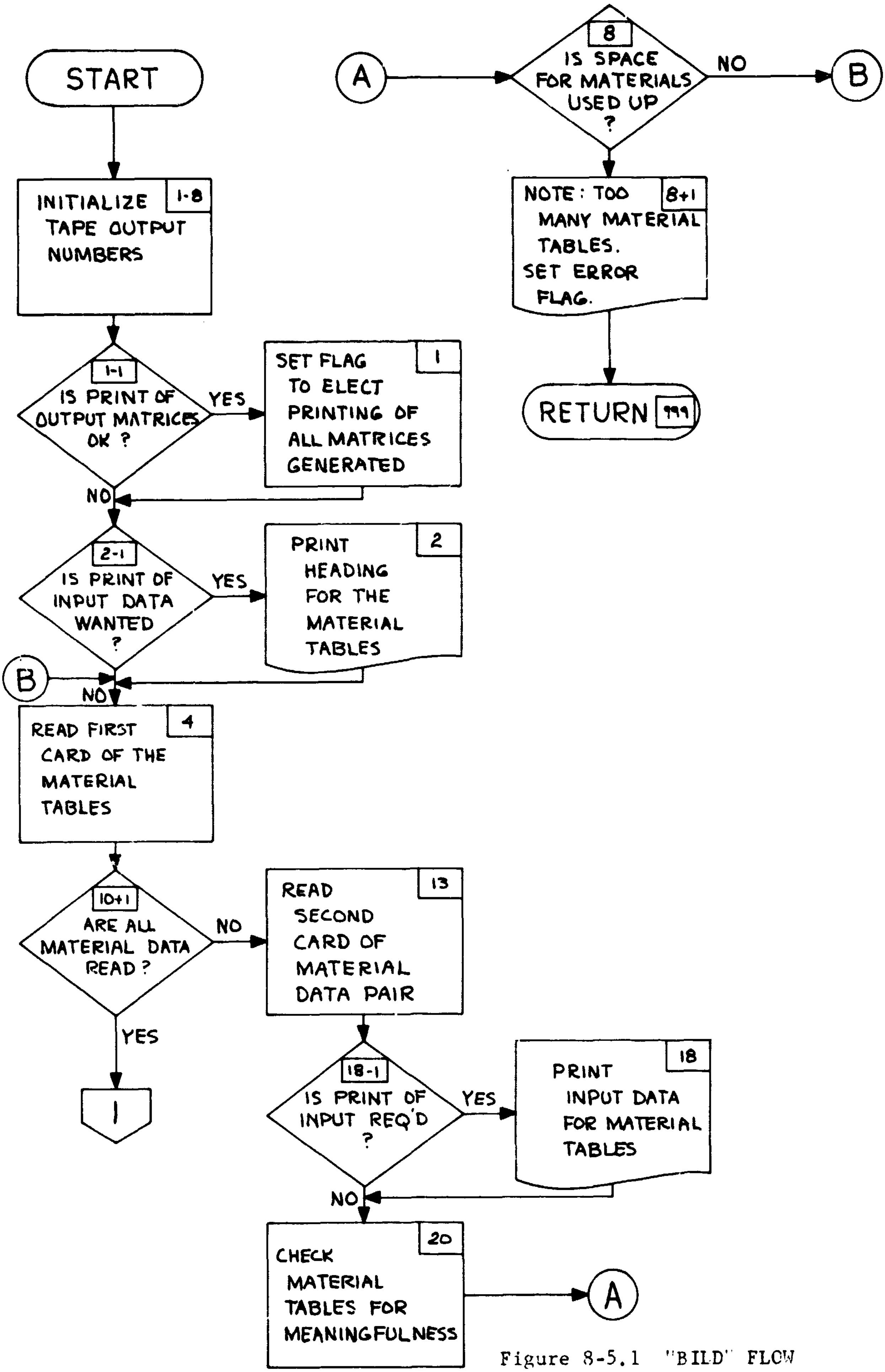



Figure 8-5.1 "BILD" FLOW

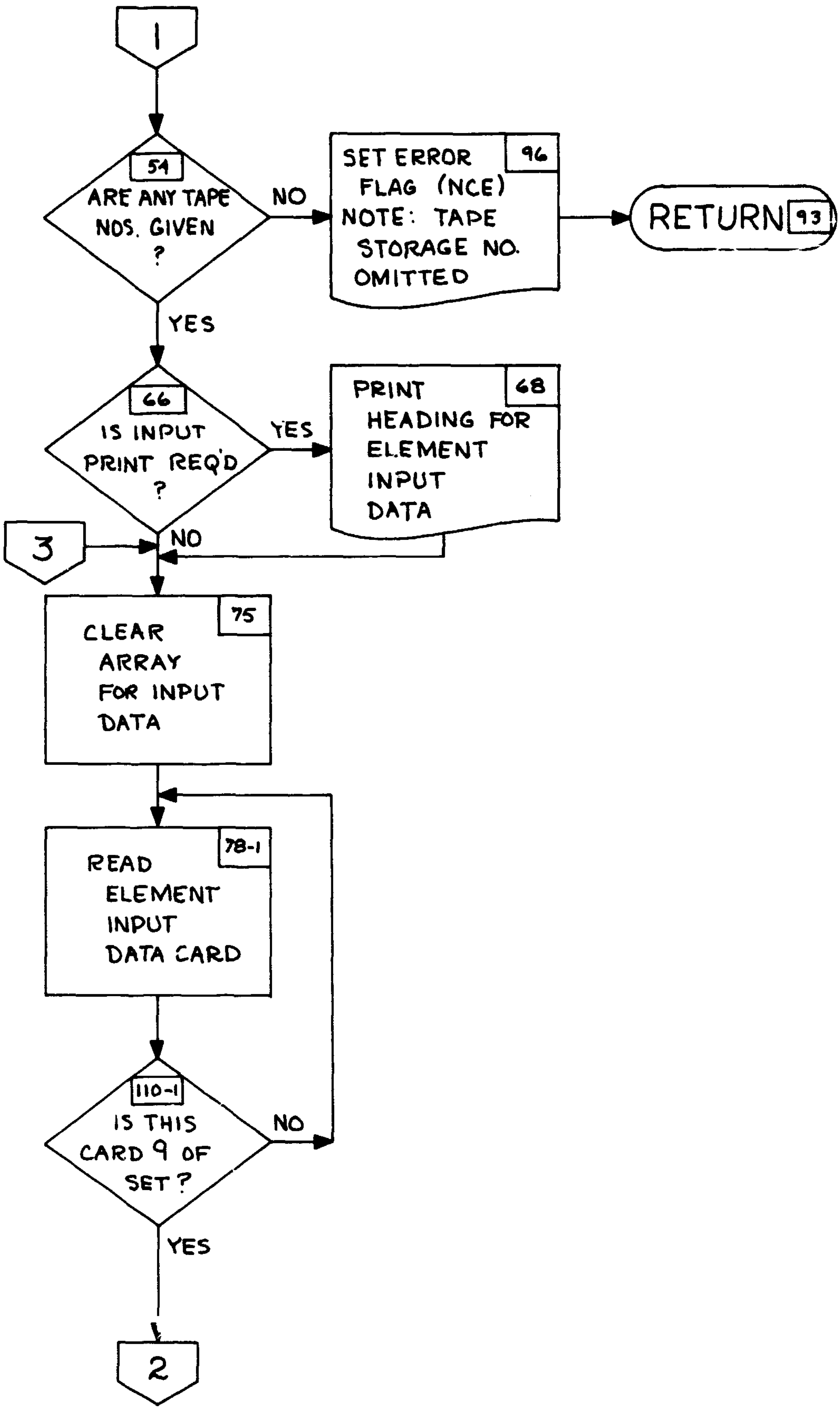



Figure 8-5.1 "BILD" FLOW

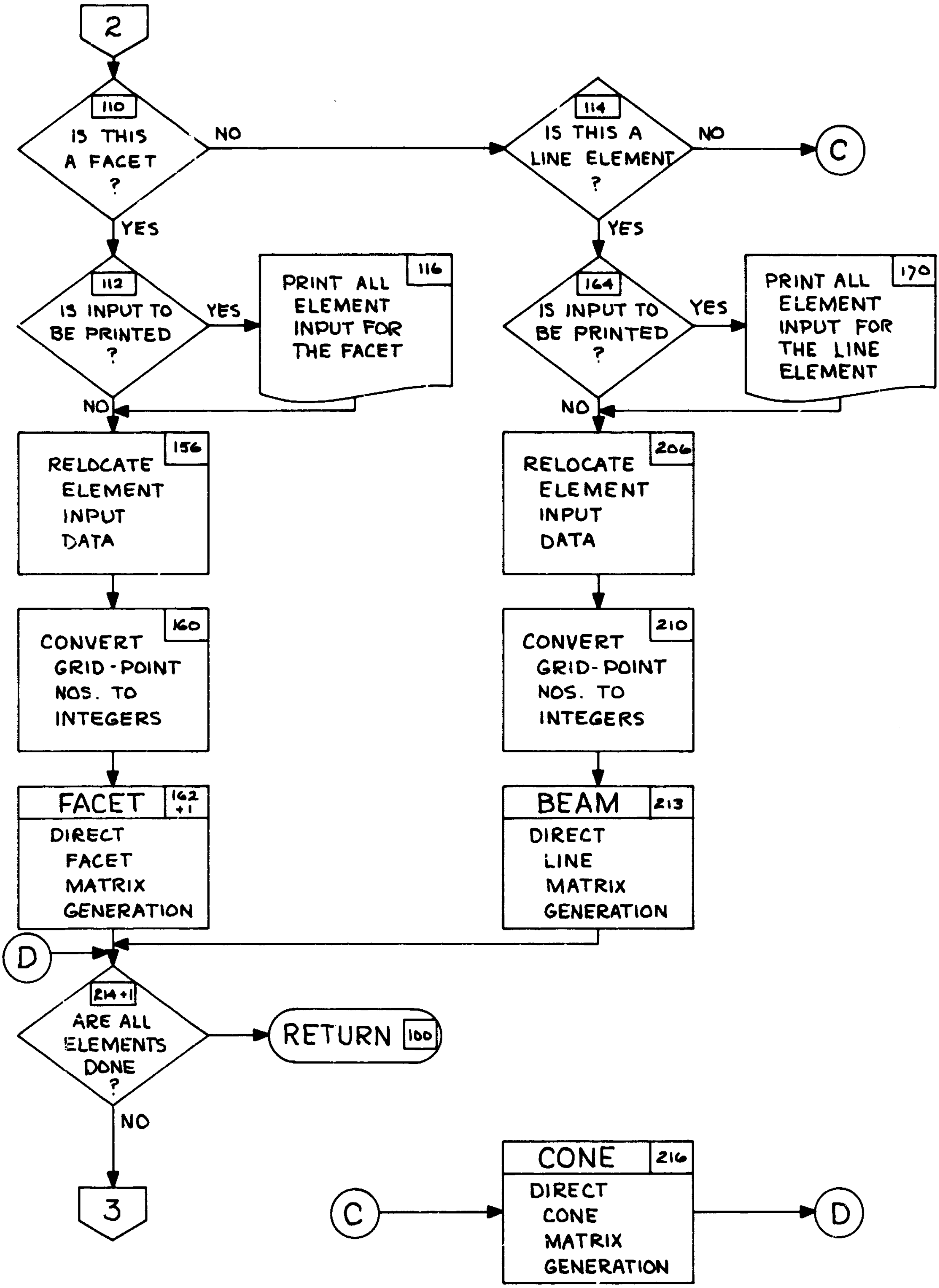



Figure 8-5.1 "BILD" FLOW

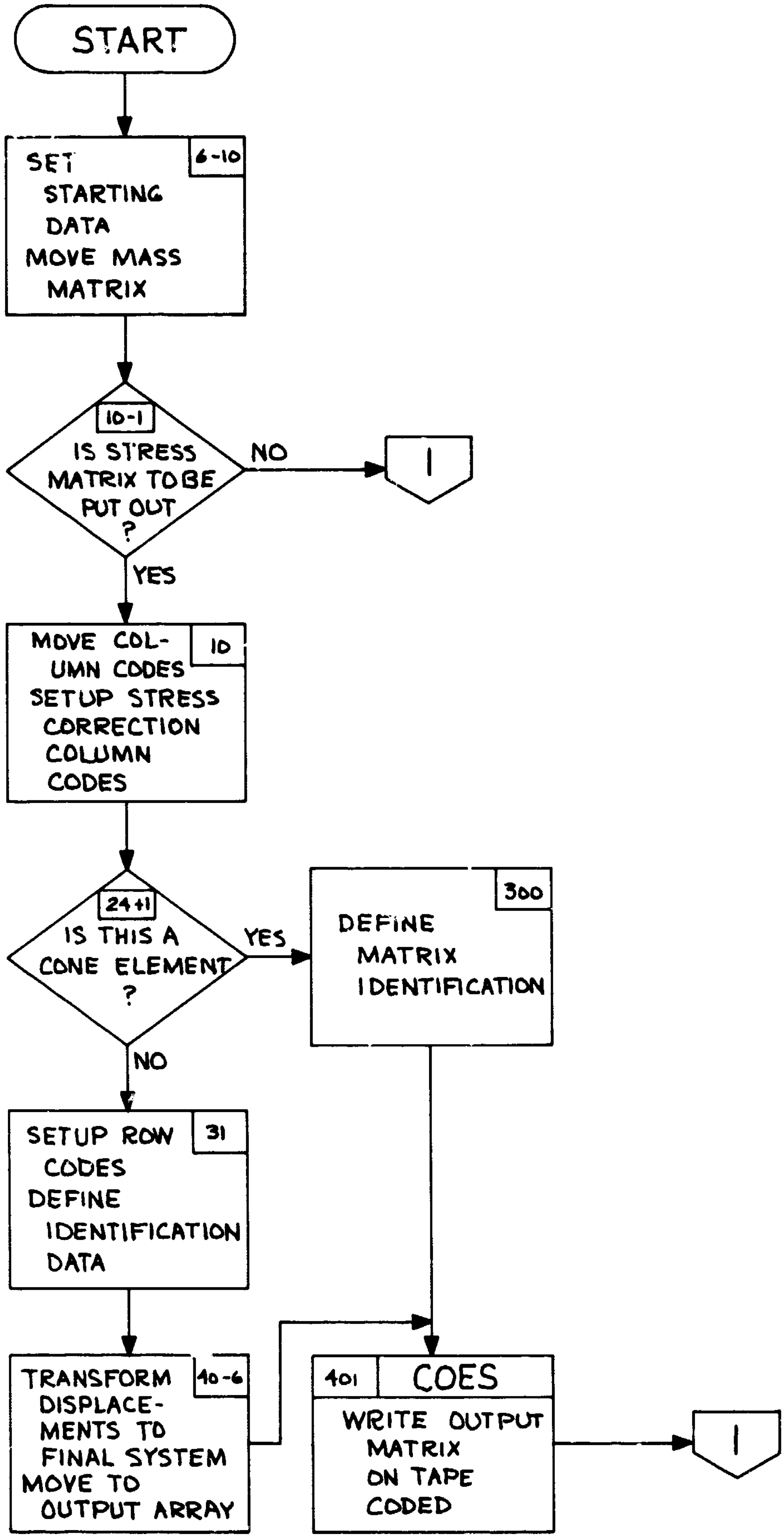



Figure 8-5.2 "SFT" FLOW

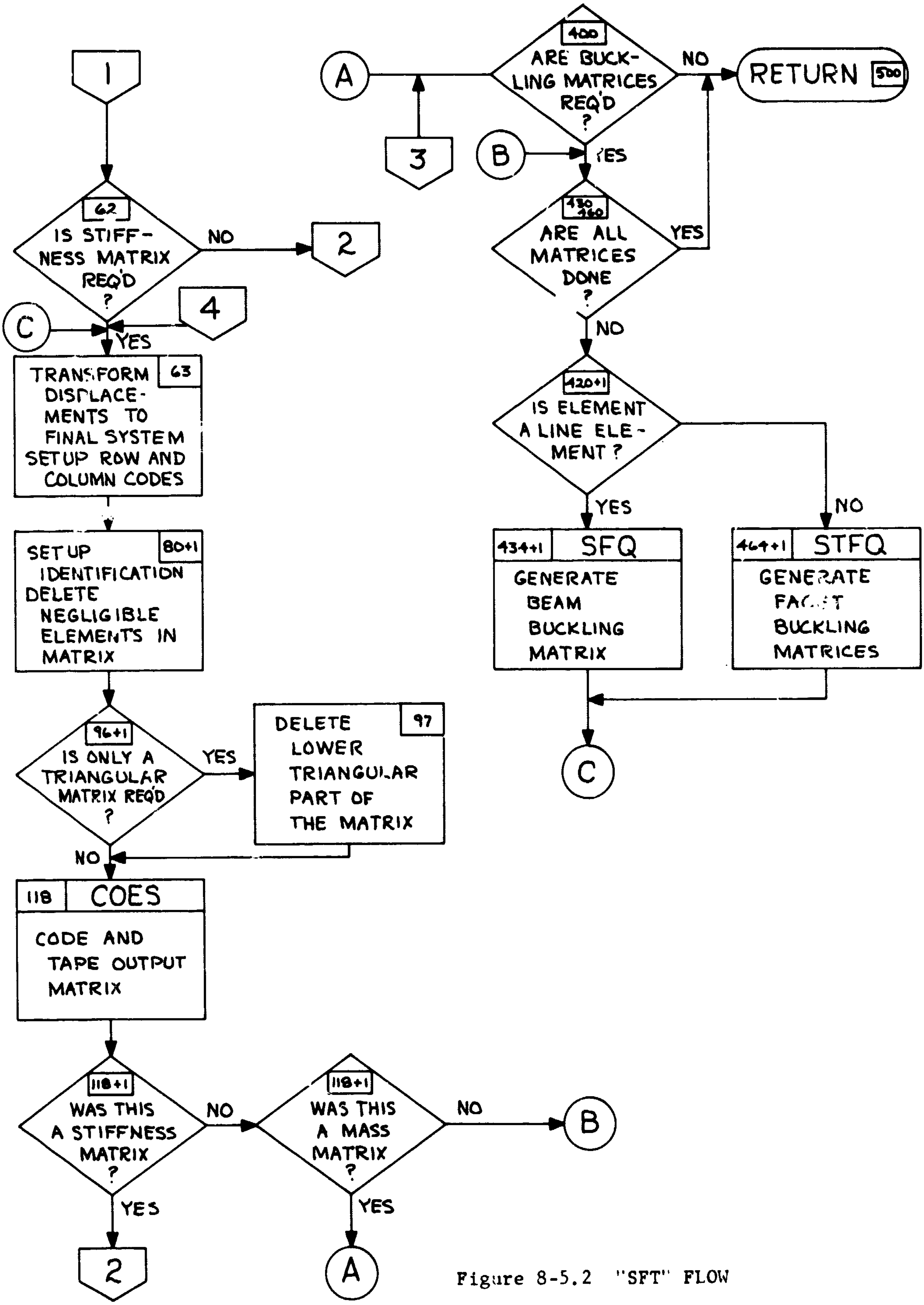



Figure 8-5.2 "SFT" FLOW

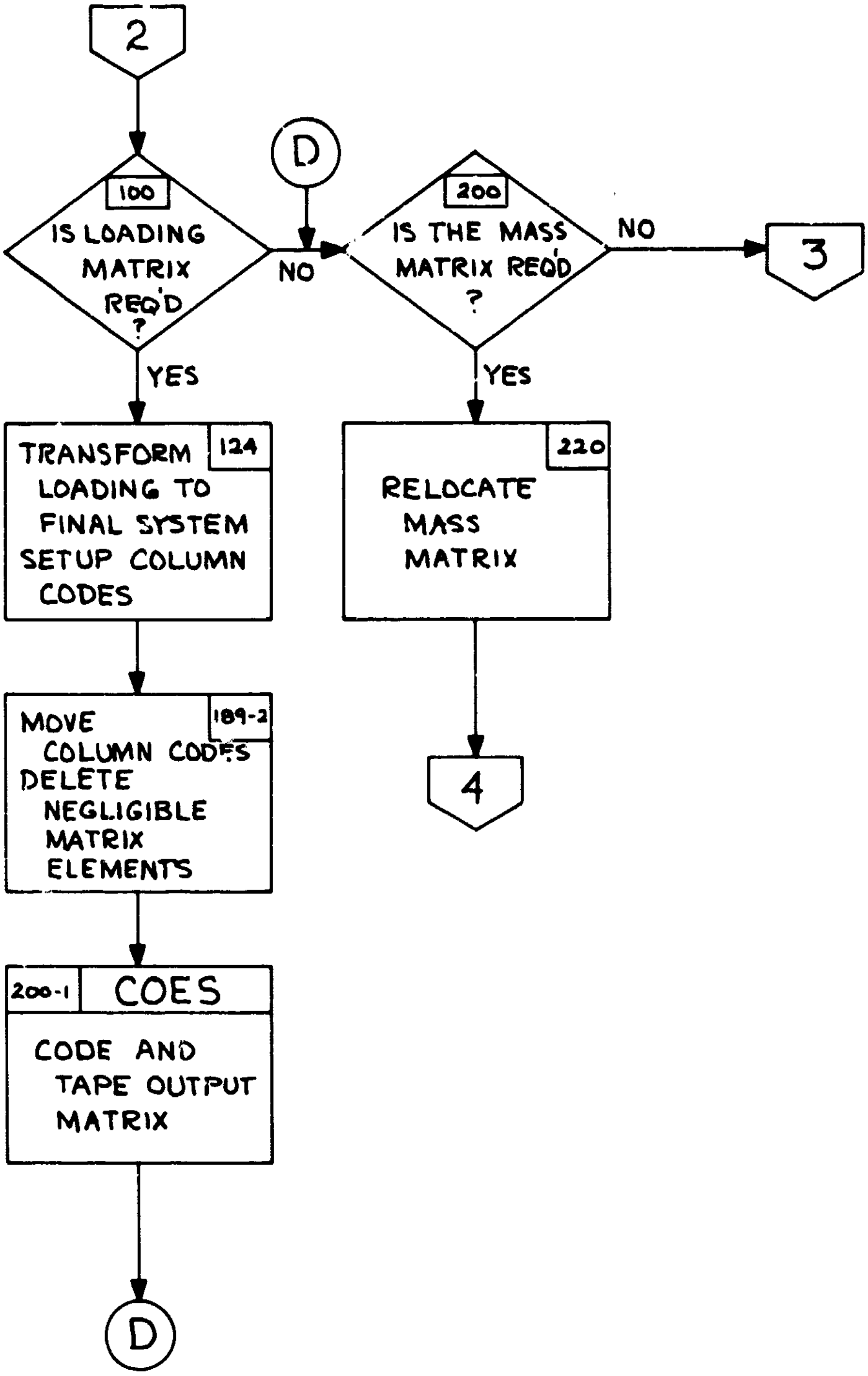



Figure 8-5.2 "SFT" FLOW

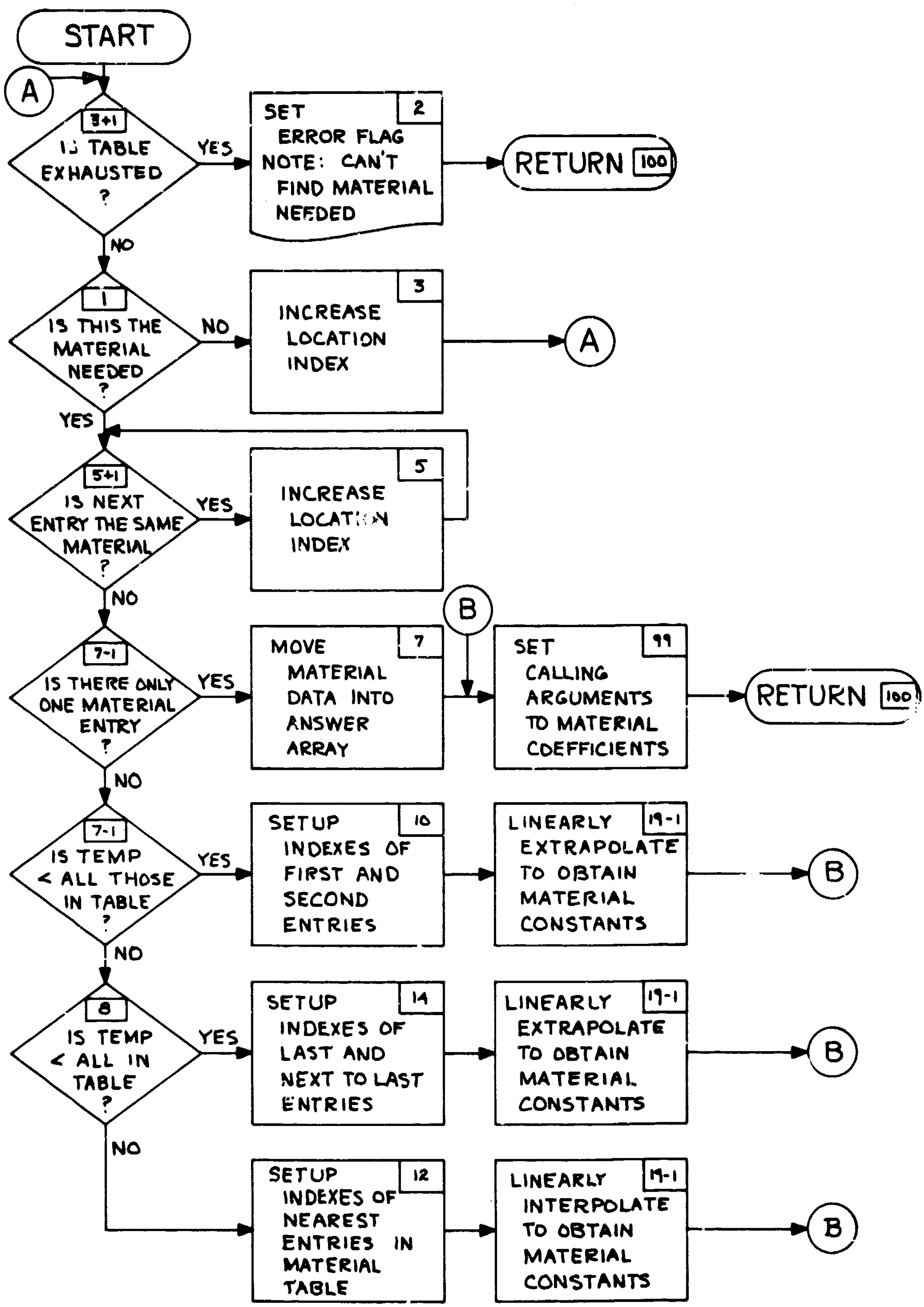



Figure 8-5.3 "TABLE" FLOW

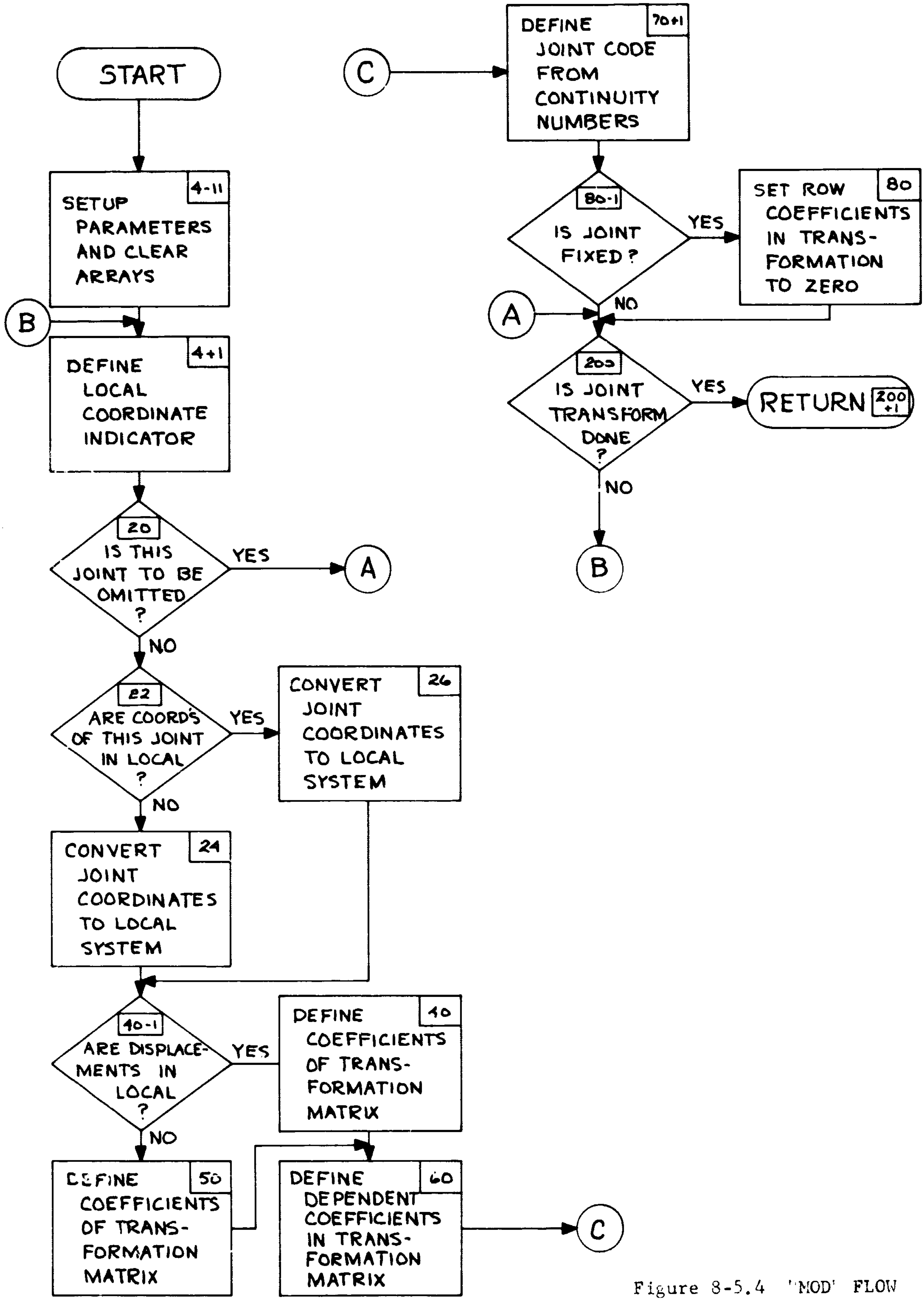



Figure 8-5.4 'MOD' FLOW

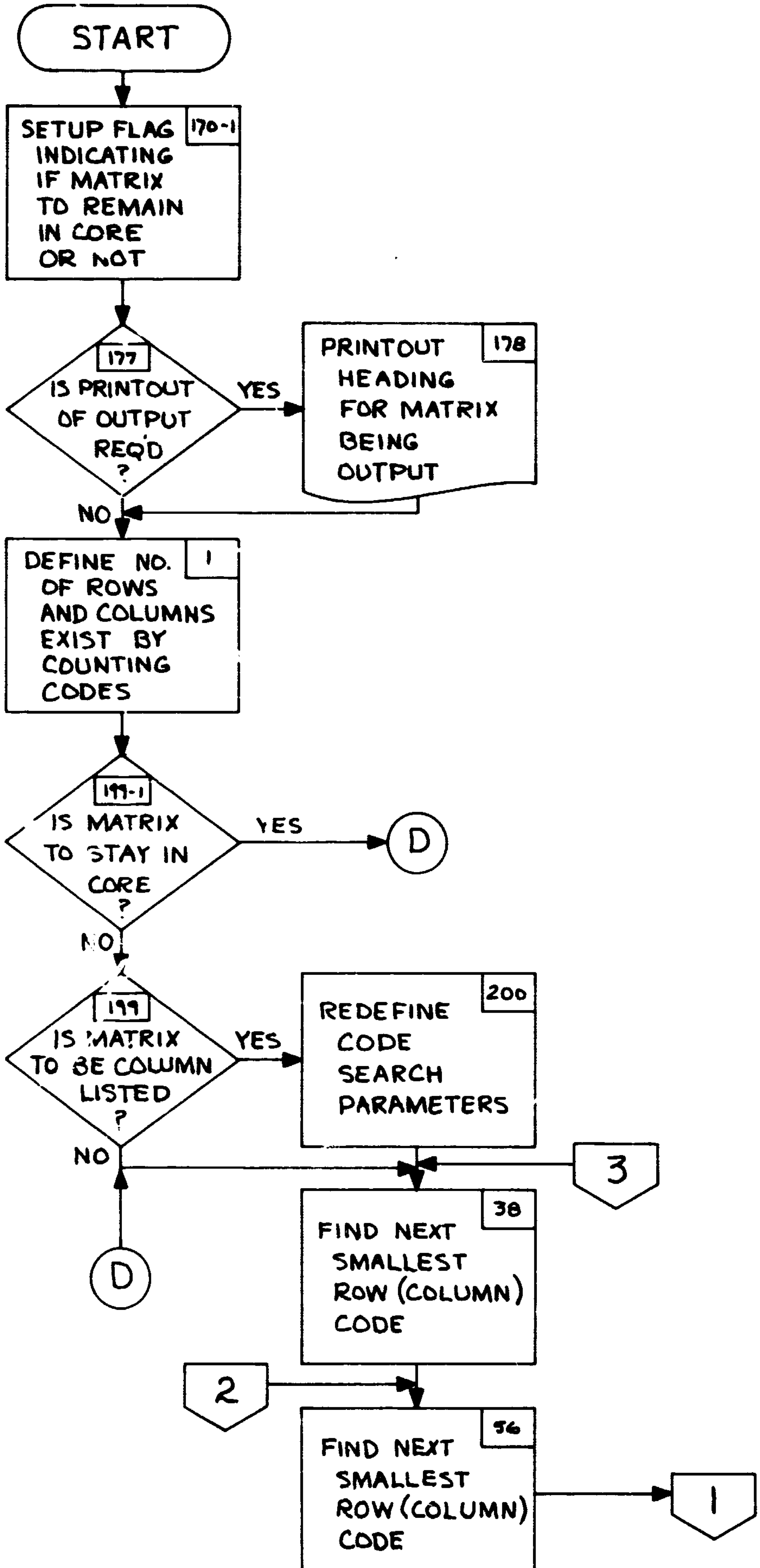



Figure 8-5.5 "COES" FLOW

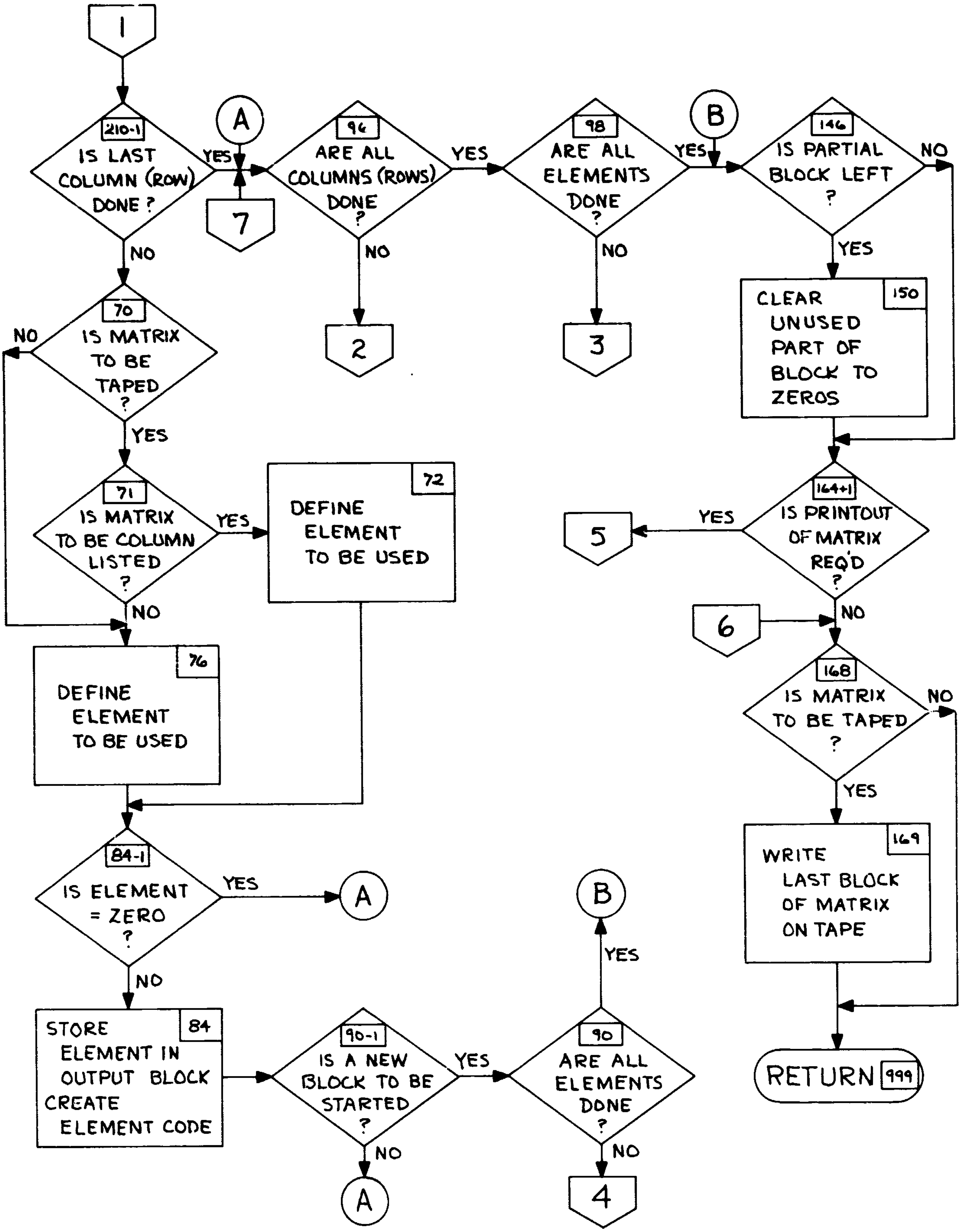



Figure 8-5.5 "COES" FLOW

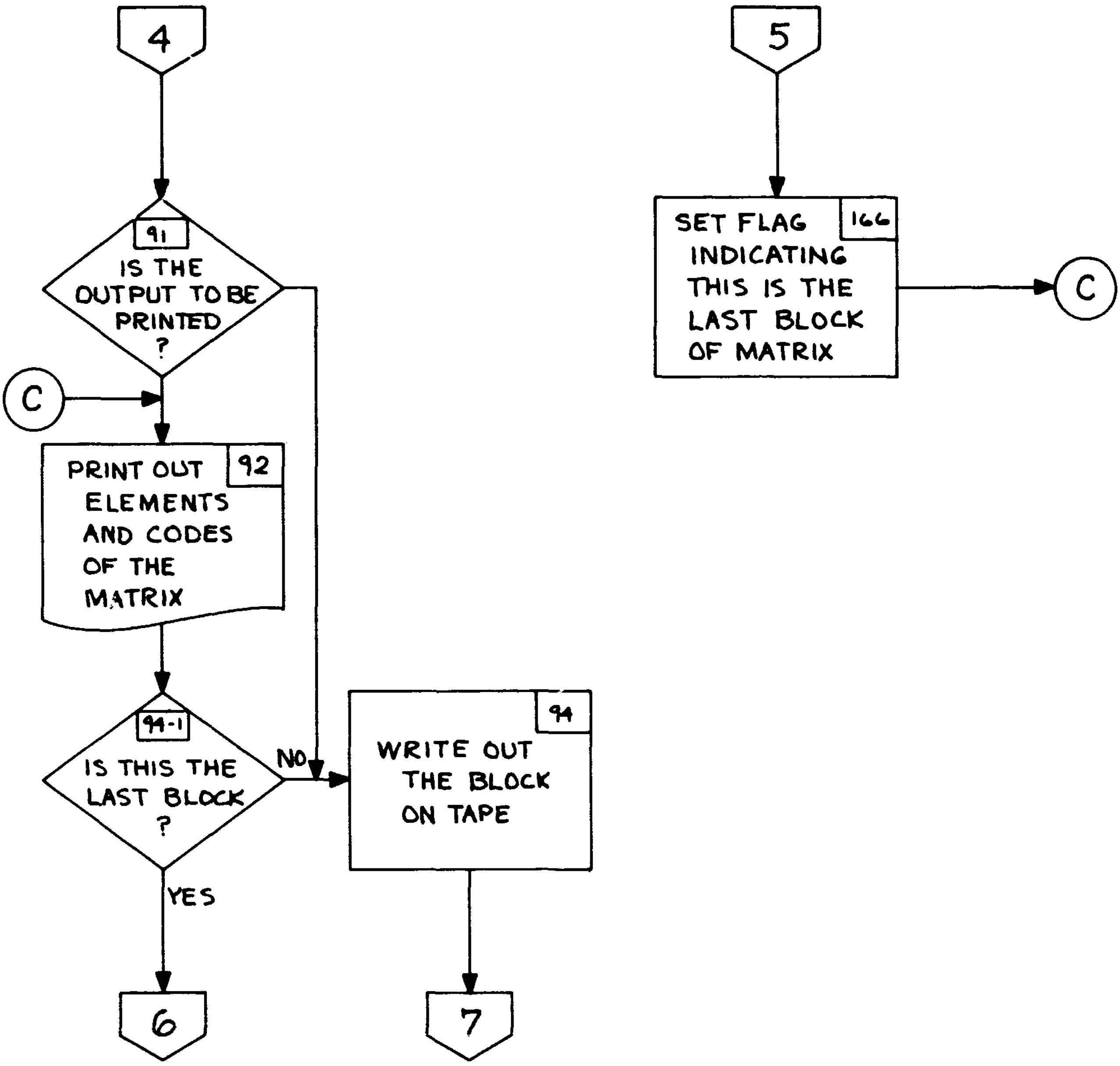



Figure 8-5.5 "COES" FLCW